



Diplomová práce

Možnosti tvorby interních aplikací malých a středních podniků integrovatelných do prostředí Microsoft Teams

Studijní program:

N2612 Elektrotechnika a informatika

Studijní obor:

Informační technologie

Autor práce:

Bc. Kristýna Frydrychová

Vedoucí práce:

Ing. Jana Vitvarová, Ph.D.

Ústav mechatroniky a technické informatiky

Liberec 2024



Diploma thesis

Development of internal applications for small and medium-sized enterprises as Microsoft Teams Apps

Study programme:

N2612 Electrical engineering and informatics

Study branch:

Information technology

Author:

Bc. Kristýna Frydrychová

Supervisor:

Ing. Jana Vitvarová, Ph.D.

Institute of Mechatronics and Computer Engineering

Liberec 2024



Zadání diplomové práce

Možnosti tvorby interních aplikací malých a středních podniků integrovatelných do prostředí Microsoft Teams

<i>Jméno a příjmení:</i>	Bc. Kristýna Frydrychová
<i>Osobní číslo:</i>	M20000167
<i>Studijní program:</i>	N2612 Elektrotechnika a informatika
<i>Studijní obor:</i>	Informační technologie
<i>Zadávající katedra:</i>	Ústav mechatroniky a technické informatiky
<i>Akademický rok:</i>	2022/2023

Zásady pro vypracování:

1. Identifikujte základní sadu interních aplikací potřebných pro řízení malých a středních podniků. Sepište jejich hlavní funkční a technické požadavky.
2. Proveďte rešerši možností tvorby a způsobů integrace interních aplikací do prostředí Microsoft Teams. Popište limity jednotlivých řešení.
3. Jako vzorovou interní aplikaci vytvořte pokročilý docházkový systém.
4. Nejdříve analyzujte existující softwarová řešení a stávající procesy týkající se docházky jako přehled dovolených, sledování času stráveného na projektech nebo plánování práce.
5. Na základě předchozí analýzy specifikujte funkční požadavky pro vzorovou aplikaci.
6. Navrhněte interaktivní prototyp uživatelského rozhraní a získejte zpětnou vazbu od potenciálních uživatelů.
7. Vyberte vhodné technické řešení a způsob integrace do prostředí Microsoft Teams.
8. Aplikaci implementujte a otestujte.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 40–50 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: čeština

Seznam odborné literatury:

- [1] VAN ROUSSELT, Rick. Pro Microsoft Teams Development: A Hands-on Guide to Building Custom Solutions for the Teams Platform. New York City: Apress, 2020. ISBN 1484263634.
- [2] NS, Jenkins. Building Solutions with Microsoft Teams. New York City: Apress, 2020. ISBN 1484264754.
- [3] BALUTE, August Anthony N., Mateo D. MACALAGUING a Dennis B. GONZALES. A Cost Effective Sharepoint Platform Attendance Tracking System. Circulation in Computer Science, 2016, 1(2), 8-13. ISSN 2456-3692. DOI: 10.22632/ccs-2016-251-32.
- [4] FIŠER, Roman. Procesní řízení pro manažery. Praha: Grada, 2014. ISBN 978-80-247-5038-5.
- [5] UI Kit pro TeamApp [online]. Figma [citace: 12. října 2022]. Dostupné z: <https://www.figma.com/community/file/916836509871353159>

Vedoucí práce: Ing. Jana Vitvarová, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce: 12. října 2022
Předpokládaný termín odevzdání: 15. května 2023

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

prof. Ing. Aleš Richter, CSc.
garant studijního programu

V Liberci dne 12. října 2022

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědoma toho, že na mou diplomovou práci se plně vztahuje zákon č.121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

14. května 2024

Bc. Kristýna Frydrychová

Poděkování

Ráda bych poděkovala paní Ing. Janě Vitvarové, Ph.D. za vstřícnost, podnětné rady, ochotu při konzultacích a odborné vedení mé diplomové práce. Velké dík také patří mému konzultantovi Michalu Čermákovi ohledně cenných rad a možnosti kombinace práce se studiem. Také bych chtěla poděkovat společnosti Cermitech, která mi umožnila práci uskutečnit poskytnutím potřebných materiálů a licencí.

Abstrakt

Cílem této diplomové práce je vytvořit efektivní docházkový systém pro malé a střední podniky, který bude integrován do prostředí Microsoft Teams. To zahrnuje představení klíčových interních aplikací nezbytných pro úspěšné řízení podniku spolu s detailní analýzou jejich funkčních a technických požadavků. Důležité je vyzkoušet různé možnosti tvorby a integrace těchto aplikací do Teams a nalézt limity jednotlivých řešení.

Dalším bodem je prozkoumání existujících softwarových řešení a pracovních postupů týkajících se docházky pro získání přehledu o jejich funkcích a případných nedostatcích. Na základě této analýzy je možné specifikovat funkční požadavky pro vzorovou aplikaci, která bude vytvořena jako součást této práce.

Návrh interaktivního prototypu uživatelského rozhraní je částí, díky které lze získat zpětnou vazbu od potenciálních uživatelů a zajistit, že aplikace bude co nejvíce vyhovovat jejich potřebám a očekáváním.

Nakonec bude potřeba se zaměřit na výběr vhodného technického řešení a způsobu integrace docházkového systému do prostředí Microsoft Teams a tuto volbu zdůvodnit. Výsledkem práce je tento zvolený způsob sestavit a integrovat a následně zhodnotit, zda aplikace splňuje veškeré požadavky a standardy kvality před uvedením do provozu.

Klíčová slova

Microsoft, MS, Teams, SharePoint, Personal Apps, Power Apps, SharePoint Framework, SPFx, Azure, App Service, ASP.NET API, SQL Database, Azure Portal, Microsoft Entra ID, MS Graph API, AadHttpClient, TypeScript, React, FluentUI, C#, T-SQL, CORS

Abstract

The focus of this thesis is to create an effective attendance system for small and medium-sized enterprises which will be integrated into the Microsoft Teams environment. This includes the introduction of key internal applications necessary for successful business management along with a detailed analysis of their functional and technical requirements. It is important to try out various options for creating and integrating these applications into Teams and to identify the limitations of each solution.

Another point is to investigate existing software solutions and workflows related to attendance to gain an overview of their features and potential shortcomings. Based on this analysis, it is possible to specify functional requirements for a prototype application that will be developed as part of this thesis.

The design of the interactive user interface prototype is the part through which feedback can be obtained from potential users and ensure that the application is as much as possible to their needs and expectations.

Finally, it will be necessary to focus on selecting a suitable technical solution and method of integrating the attendance system into the Microsoft Teams environment and justify this choice. The result of this thesis is to build and integrate this chosen method and then evaluate whether the application meets all requirements and quality standards before deployment.

Keywords

Microsoft, MS, Teams, SharePoint, Personal Apps, Power Apps, SharePoint Framework, SPFx, Azure, App Service, ASP.NET API, SQL Database, Azure Portal, Microsoft Entra ID, MS Graph API, AadHttpClient, TypeScript, React, FluentUI, C#, T-SQL, CORS

Obsah

Seznam obrázků	12
Seznam tabulek	13
Seznam zdrojových kódů	13
Seznam zkratk	14
Úvod	15
1 Aplikace pro řízení menších podniků	16
1.1 Komunikace s pracovním týmem a se zákazníky	17
1.2 Plánování směn a automatizace pracovních postupů	18
1.3 Správa, řízení a plánování projektů	18
1.4 Výkaz práce, rozpočty a účetnictví	19
1.5 Uložiště dat	20
1.6 Prezentace společnosti	21
2 Tvorba a integrace aplikací do prostředí Microsoft Teams	22
2.1 Možnosti tvorby aplikací pro MS Teams	22
2.2 Způsob integrace aplikací do prostředí Microsoft Teams	23
2.3 Zmapování funkcí pro rozšíření vlastností Microsoft Teams	24
2.4 Osobní aplikace (Personal App)	25
2.4.1 Power Apps	25
2.4.1.1 Tvorba aplikací přes Power Apps	26
2.4.1.2 Integrace Power Apps aplikací do MS Teams	28
2.4.1.3 Výhody a nevýhody Power Apps	28
2.4.1.4 Limity low-code programování	30
2.4.2 SharePoint Framework (SPFx)	33
2.4.2.1 Tvorba aplikací pomocí SPFx	33
2.4.2.2 Způsob integrace SPFx aplikací do MS Teams	34
2.4.2.3 Výhody a nevýhody SPFx	35
2.4.3 Existující webová aplikace	38
2.4.3.1 Tvorba webové aplikace	38
2.4.3.2 Způsob integrace existující webové aplikace do MS Teams	39
2.4.3.3 Výhody a nevýhody existující webové aplikace	42
2.4.4 Srovnání způsobů implementace aplikací do MS Teams	45

3	Docházkové systémy a stávající procesy pro vedení docházky v MS Teams	48
3.1	Existující systémy pro vedení docházky	48
3.1.1	Srovnání existujících systémů pro vedení docházky.....	51
3.2	Stávající procesy pro vedení docházky v Microsoft 365.....	51
3.3	Důvody vytvoření pokročilého docházkového systému.....	52
3.3.1	Získání přehledu pro mzdy.....	53
3.3.2	Vygenerování výkazu práce pro zákazníka.....	54
3.3.3	Interní sledování nákladů projektů	55
3.3.4	Vygenerování účetní rozpracovanosti	55
4	Funkční požadavky pro docházkový systém	57
5	Návrh aplikace docházkového systému	59
5.1	Datová vrstva	59
5.2	Uživatelské rozhraní a fungování aplikace.....	62
5.2.1	Docházka	62
5.2.2	Statistiky	66
5.2.3	Management	68
5.2.4	Export	69
5.2.5	Nastavení	71
6	Tvorba a integrace SPFx aplikace.....	73
6.1	SharePoint Framework Extensions.....	73
6.2	Příprava prostředí	74
6.3	Vytvoření základu SPFx aplikace spolu s FluentUI.....	75
6.4	Vytvoření manifestu	76
6.5	Nahrání na SharePoint	77
6.6	Testování	77
6.6.1	Lokálně přes Workbench	78
6.6.2	Online přes Workbench.....	78
6.6.3	Lokálně přes Teams	78
6.6.4	Online přes Teams.....	79
6.7	Konfigurace a další úpravy v aplikaci	79
6.8	Práce s daty	80
6.8.1	Propojení se SharePoint pomocí SPHttpClient	80
6.8.2	Propojení s Microsoft 365 pomocí MSGraphClient	81

6.8.2.1	Vytvoření oprávnění SPFx ke službám Microsoft 365	82
6.8.2.2	Práce s MS Graph API v aplikaci SPFx.....	83
6.8.3	Propojení s databází Azure pomocí AadHttpClient	84
6.8.3.1	Vytvoření SQL databáze a SQL serveru v Azure Portal	85
6.8.3.2	Vytvoření webového rozhraní API ASP.NET Core	86
6.8.3.3	Komunikace API s databází pomocí SqlConnection.....	87
6.8.3.4	Publikování webového rozhraní API ASP.NET Core do Azure App Service	89
6.8.3.5	Konfigurace zabezpečeného připojení mezi službou App Service a databází SQL pomocí spravované identity	89
6.8.3.6	Registrování aplikace v Microsoft Entra ID	91
6.8.3.7	Nastavení CORS pro přístup ke zdrojům v jiné doméně	92
6.8.3.8	Komunikace SPFx s webovým rozhraním pomocí AadHttpClient	93
6.8.3.9	Nastavení zabezpečeného přístupu k API pomocí API Scope.....	94
6.8.3.10	Průběh zabezpečené komunikace mezi klientskou částí SPFx a API ve službě Azure App Service	95
7	Testování docházky	97
	Závěr.....	99
	Citovaná literatura	100
	Příloha A	102
	Příloha B.....	104

Seznam obrázků

Obrázek 1: Diagram funkcí pro rozšíření vlastností MS Teams	24
Obrázek 2: Use case postupu tvorby aplikace přes Power Apps	26
Obrázek 3: Náhled aplikace vytvořené v Power Apps a publikované v MS Teams.....	28
Obrázek 4: Use case postupu tvorby aplikace pomocí SPFx	34
Obrázek 5: Use case postupu integrace SPFx aplikace do MS Teams	35
Obrázek 6: Use case postupu integrace existující webové aplikace do MS Teams	39
Obrázek 7: Přehled fungování docházky v souboru MS Excel.....	52
Obrázek 8: Postup pro získání přehledu pro mzdy z docházky	53
Obrázek 9: Postup pro vygenerování výkazu práce pro zákazníka.....	54
Obrázek 10: Postup pro získání interní sledovanosti nákladů projektu	55
Obrázek 11: Postup pro vygenerování účetní rozpracovanosti	56
Obrázek 12: Diagram přehledu oprávnění pro uživatele a správce	57
Obrázek 13: ER diagram docházkového systému	60
Obrázek 14: Záložka "Docházka" pro záznam docházky přihlášeného uživatele	63
Obrázek 15: Vývojový diagram pro záznam práce na projektech pomocí tlačítek v docházce	65
Obrázek 16: Záložka "Statistiky" pro zobrazení přehledů pro uživatele	68
Obrázek 17: Záložka "Management" umožňuje správu docházky uživatelů správcem.....	69
Obrázek 18: Záložka "Export" pro generování dokumentů na základě dat z databáze	70
Obrázek 19: Záložka "Nastavení" pro základní nastavení a práci s databází	72
Obrázek 20: Vygenerování SPFx aplikace pomocí Yeoman generátoru	75
Obrázek 21: Architektura aplikace.....	84
Obrázek 22: Zabezpečená komunikace mezi službou App Service a SQL databází přes Microsoft.Data.SqlClient díky automatickému ověřování přes Microsoft Entra ID	91
Obrázek 23: Proces propojení webové části na straně klienta SPFx s autentizovaným Azure API.....	95

Seznam tabulek

Tabulka 1: Potřeby pro řízení podniku a příklady aplikací, převážně z balíčku Microsoft 365, pro danou činnost využívané.....	16
Tabulka 2: Výhody a nevýhody tvorby aplikací v Power Apps	29
Tabulka 3: Limity tvorby low-code aplikací, kterými jsou aplikace v Power Apps.....	31
Tabulka 4: Výhody a nevýhody použití SPFx pro tvorbu aplikací	36
Tabulka 5: Výhody a nevýhody použití existující webové aplikace.....	43
Tabulka 6: Srovnání vlastností pro tvorbu aplikací do MS Teams v Power Apps, pomocí SPFx a vložením existující webové aplikace	46
Tabulka 7: Srovnání vybraných existujících systémů pro vedení docházky	51
Tabulka 8: Funkční požadavky na docházku	58
Tabulka 9: Přehled zobrazovaných statistik pro zaměstnance	66

Seznam zdrojových kódů

Zdrojový kód 1: HTML soubor s tagem iFrame	40
Zdrojový kód 2: HTML soubor s „Execution Deeplink“ na kartu v kanálu Teams	41
Zdrojový kód 3: Úprava konfiguračního souboru config/serve.json	79
Zdrojový kód 4: Získání názvu SP webu za použití instance SPHttpClient	80
Zdrojový kód 5: Získání položek seznamu z SP webu pomocí knihovny PnPjs	81
Zdrojový kód 6: Oprávnění SPFx aplikace pro čtení základních profilů všech uživatelů v souboru package-solution.json	82
Zdrojový kód 7: Příprava prostředí pro komunikaci s Microsoft Graph API z aplikace SPFx	83
Zdrojový kód 8: Připojovací řetězec bez hesla pro připojení ASP.NET API k Azure SQL Database	87
Zdrojový kód 9: Připojení ke službě Azure SQL Database z API pomocí Microsoft.Data.SqlClient	88
Zdrojový kód 10: Připojení SPFx k rozhraní API v App Service pomocí AadHttpClient	94
Zdrojový kód 11: Zabezpečený přístup k API pouze přihlášeným uživatelům nebo SPFx aplikaci	94

Seznam zkratek

MS 365 – Microsoft 365 – sada podnikových aplikací a služeb od společnosti Microsoft

SP – SharePoint – webová platforma od MS pro správu a ukládání dokumentů, sdílení informací prostřednictvím intranetu, implementaci interních aplikací a obchodních procesů

SPFx – SharePoint Framework Extensions – model pro rozšiřování služeb Microsoft 365 zahrnutím komponent řízených JavaScriptem do uživatelského rozhraní

UI – User Interface – uživatelské rozhraní k interakci mezi člověkem a programem

HTTP – Hypertext Transfer Protocol – protokol pro komunikaci mezi serverem a prohlížečem

HTTPS – Hypertext Transfer Protocol Secure – internetový protokol umožňující zabezpečenou komunikaci v počítačové síti pomocí šifrování a integrity dat

REST – Representational State Transfer – architektura pro odesílání požadavků pro načtení nebo úpravu zdrojů na serveru pomocí jednoduchých HTTP volání

API – Application Programming Interface – webové rozhraní, které pro komunikaci mezi dvěma a více platformami pro výměnu dat využívá REST

URI – Uniform Resource Identifier – jednotný identifikátor pro specifikaci zdroje informací

URL – Uniform Resource Locator – webová adresa pro specifikaci umístění zdroje informací

HTML – Hypertext Markup Language – značkovací jazyk pro tvorbu webových stránek

CSS – Cascading Style Sheets – jazyk pro způsob zobrazení elementů na HTML stránkách

ERD – Entity Relationship Diagram – metoda pro vytvoření schémat datových modelů systému

SQL – Structured Query Language – strukturovaný dotazovací jazyk pro práci s daty v relačních databázích

DOM – Document Object Model – struktura a obsah dokumentu tvořený datovou reprezentací objektů, ke kterým mohou přistupovat skripty a měnit tak jejich strukturu, styl nebo obsah

SDK – Software Development Kit – sada nástrojů pro vývoj softwaru pro specifickou platformu

NPM – Node Package Manager – správce javascriptových balíčků pro jejich instalaci a údržbu

JS – JavaScript – multiplatformní, interpretovaný, objektově orientovaný skriptovací jazyk nejčastěji využívaný při tvorbě webových stránek

JSON – JavaScript Object Notation – způsob zápisu dat určený pro přenos dat

PDF – Portable Document Format – souborový formát pro ukládání a zobrazení dokumentu nezávisle na zařízení

CORS – Cross-Origin Resource Sharing – mechanismus pro přístup ke zdroji dat webovým aplikacím, které běží na jiné doméně, pomocí hlaviček protokolu HTTP

GUID – Globally Unique Identifier – univerzální identifikátor v Microsoft systémech

Azure AD – Azure Active Directory (Microsoft Entra ID) – služba pro správu identit a přístupu

VS – Visual Studio – integrované vývojové prostředí (IDE) od společnosti Microsoft

Úvod

S rostoucí dostupností a snadným využitím moderních technologií se i menší podniky snaží využít možností automatizace a vlastního přizpůsobení si pracovního postupů na míru k zefektivnění svých interních procesů. Organizace obvykle využívají různé sady balíčků, které obsahují produkty a služby k celkovému řízení a koordinaci svého podniku. Jedním takovým balíčkem může být i Microsoft 365. Ten kromě sady aplikací, cloudových služeb a vestavného zabezpečení nabízí také možnosti rozšíření platformy MS Teams o vlastní automatizované procesy, drobné karty nebo osobní aplikace. Díky tomu může uživatel přistupovat ke všem potřebným službám z jednoho prostředí bez nutnosti stálého přepínání kontextu mezi službami a zároveň snadno pracovat se zdrojem dat z dalších integrovaných aplikací.

Cílem této diplomové práce je prozkoumat způsoby tvorby a integrace aplikací do prostředí Microsoft Teams, jednotlivé možnosti porovnat a následně vytvořit vzorovou interní aplikaci na základě zvoleného technického řešení. Práce se zaměřuje na návrh pokročilého docházkového systému pro malé a střední podniky podle zadaných kritérií, vývoj této aplikace a její následnou integraci na platformu Teams.

První část bude zaměřená na identifikaci hlavních funkčních a technických požadavků interních aplikací nezbytných pro efektivní spolupráci a řízení podniku.

Následující fáze se bude věnovat prozkoumání možností rozšiřitelnosti platformy Microsoft Teams, a to především o osobní aplikace. To zahrnuje rozbor postupů tvorby těchto aplikací, jejich integrace a popsání případných limitů každého řešení.

V průběhu práce budou analyzovány již existující softwarová řešení a procesy týkající se docházky a také vytvořeny návrhy interaktivního prototypu uživatelského rozhraní na základě specifikovaných funkčních požadavků pro vzorovou aplikaci.

Dalším krokem bude volba vhodného technického řešení a způsobu integrace aplikace na platformu Teams. To také zahrnuje detailní popis implementace aplikace ve zvoleném technickém řešení.

Nakonec budou popsány vlastnosti výsledné aplikace a zhodnocen přínos docházkového systému integrovaného do prostředí Microsoft Teams k efektivnějšímu řízení podniku.

1 Aplikace pro řízení menších podniků

Sady interních aplikací nabízí podnikům širokou škálu nástrojů pro efektivní řízení a automatizaci podnikových procesů a k zajištění bezpečnosti a ochrany dat. Jejich používání zlepšuje produktivitu a spolupráci týmu na projektech. Příkladem těchto aplikací určených pro efektivní podnikové řízení je sada produktů a služeb v balíčku Microsoft 365. [23]

Tabulka 1: Potřeby pro řízení podniku a příklady aplikací, převážně z balíčku Microsoft 365, pro danou činnost využívané

Potřeby pro řízení podniku	Aplikace k dané činnosti
Interní komunikace	Microsoft Teams Chat
Komunikace se zákazníky	Microsoft Outlook
Plánování směn a přehled vytíženosti zaměstnanců	Microsoft Teams Shifts
Automatizace pracovních postupů	Microsoft Power Automate
Správa projektů a jejich řízení	Microsoft Teams Team / MS Project
Plánování a úkoly k projektům	Microsoft Planner
Poznámky a reporty k projektům	Microsoft OneNote
Rozpočet a správa zakázek	Microsoft Excel
Docházka a výkaz hodin	Microsoft Excel
Analýza dat	Microsoft Power BI
Účetnictví a finanční správa	Altus Vario / Abra
Uložiště sdílených souborů	Microsoft SharePoint
Záloha dat	Microsoft OneDrive
Repozitář a git	Azure DevOps
Servery	Azure Portal
Web	Wedos

1.1 Komunikace s pracovním týmem a se zákazníky

At' už se jedná o domluvení termínu schůzky nebo řešení problematiky u projektu, je součástí řízení podniků i aplikace pro rychlou a efektivní interní komunikaci s týmem, ale také aplikace pro externí komunikaci se zákazníky.

Microsoft Teams

Platforma Microsoft Teams (označována také jako MS Teams nebo pouze Teams) je prostředí pro komunikaci a spolupráci na projektech. Pro získání rychlejších přehledů integruje spolu s dalšími aplikacemi a službami v rámci Microsoft 365. Vývojářům umožňuje integraci jejich vlastních aplikací a služeb přímo do platformy za účelem snížení přepínání kontextu, usnadnění spolupráce a zvýšení produktivity. Nejčastěji využívané aplikace v prostředí Microsoft Teams jsou Chat pro interní komunikaci, Týmy pro řízení projektů, Směny pro plánování a přehled vytíženosti lidí a Notifikace pro stálou informovanost o důležitých událostech, schůzkách nebo zprávách.

- Chat – uživatelé mohou komunikovat mezi sebou, vytvářet skupiny, sdílet příspěvky a dokumenty a nahrávat přílohy přímo do konverzací. Zahrnuje také volání, videohovory a sdílení obrazovky. Zaměřuje se především na interní komunikaci, ale pokud zákazník také využívá Teams, je možné přes Teams vést i externí komunikaci.
- Notifikace – upozorní zaměstnance na označení v události, která vyžaduje jeho pozornost. Z notifikace je uživatel odkázán přímo k příslušné činnosti.
- Kalendář – je propojený s diářem v Outlooku, díky čemuž zaměstnanec získává přehled své vytíženosti pro rychlé plánování dalších schůzek nebo směn.
- Aplikace – lze vybírat z již vytvořených aplikací zveřejněných ostatními organizacemi nebo si vytvořit a následně integrovat vlastní aplikace pro všechny členy organizace.

Microsoft Outlook

Microsoft Outlook zahrnuje klienta elektronické pošty, správce kontaktů, nástroje pro organizaci času a práce (kalendář a úkoly), poznámky a deník. Je tak hlavním mostem mezi zákazníkem a poskytovatelem. Pomocí kombinace jednotlivých modulů (pošta, kalendář, úkoly) je možné efektivně organizovat spolupráci s kolegy i zákazníky jak z pohledu času (navrhování schůzek, využívání sdílených kalendářů, vytváření schůzek přes Teams, připojování poznámkových bloků OneNote ke schůzkám), tak i z pohledu činností (plánování a přidělování úkolů, sdílení dokumentů a další). [7]

1.2 Plánování směn a automatizace pracovních postupů

Základem pro efektivní fungování provozu jsou dobře naplánované směny nejen pro zaměstnance, ale i pro manažery, kteří tak získávají přehled o tom, na jakých projektech se aktuálně pracuje a jaký zaměstnanec má volnou kapacitu pro přiřazení k projektu.

Microsoft Teams – Směny

Aplikace Směny na platformě Teams je určena jako kalendářní přehled směn uživatelů a jejich dovolených či nemocenských, což monitoruje vyčerpání jednotlivých lidí k plánování práce na projektech celého týmu. Umožňuje žádat o volno, o nabídku směn či prohození směn.

Microsoft Power Automate

Power Automate umožňuje uživatelům vytvářet automatizované pracovní postupy (workflows) a integraci mezi různými aplikacemi a službami. Power Automate poskytuje uživatelům nástroje pro snadné vytváření automatizovaných toků práce, které propojují různé aplikace a systémy, a tím usnadňují a urychlují spoustu běžných pracovních procesů, například:

- Toky mezi aplikacemi – například propojení e-mailových zpráv se systémem pro zákaznický servis.
- Toky v reálném čase – například sledování sociálních sítí a automatické přidávání nových příspěvků do interního chatu.
- Toky pro vytvoření formulářů a dotazníků – například automatické generování a odesílání e-mailů s dotazníky a sledování odpovědí.
- Toky pro řízení projektů – například automatické vytváření úkolů v systému pro správu projektů při vytvoření nové zakázky.

Power Automate integruje s mnoha dalšími aplikacemi a službami, jako Microsoft Teams, SharePoint, Outlook, OneNote, Planner, OneDrive, Power BI, Excel, různé databáze atd. Vytváření toků probíhá formou skládání bloků, které vykonávají konkrétní funkce, a vyžaduje minimální znalosti programování – jedná se o „low-code“ platformu.

1.3 Správa, řízení a plánování projektů

Nástroje pro řízení projektů pomáhají týmům lépe organizovat a spravovat projekty. Ať už se jedná o seznamy nebo časové nástěnky, je cílem poskytnout takovou platformu, ve které budou mít všichni přehled o každém projektu.

Microsoft Teams – Týmy

Týmy jsou součástí platformy Teams a vytváří přehled projektů a týmů. Zahrnují příspěvky a komentáře k projektům a rychlý přístup k souborům na webu SharePoint, ale i k dalším službám, jako OneNote, kde se vedou poznámky, postupy a denní práce na projektech, nebo Planner s přehledem úkolů k projektům. Pomocí seznamů na SharePointu lze vytvářet i vlastní tabulkový seznam s přehledem projektů, který lze doplnit i o toky vytvořené v Power Automate, a tím automatizovat některé činnosti při vytvoření nového projektu či změně již existujícího projektu.

Microsoft Project

Aplikace pro řízení projektů Microsoft Project nabízí přehled vytvořených projektů a jejich milníků v různých rozložení, včetně v Ganttova diagramu, pro naplánování a sledování posloupnosti činností na projektech v čase. Ovšem není ve standardním balíčku a je tedy licencovaná.

Microsoft Planner

V Planneru lze vytvářet celé plány rozdělené na kontejnery a jednotlivé úkoly i podúkoly a přiřazovat je zaměstnancům. S rostoucím počtem úkolů je vytvářen přehled s bohatým obsahem všech úkolů. Kromě přiřazení uživatelů lze zadat i termíny a prioritu úkolů, poznámky, podúkoly či dokončení úkolu.

Microsoft OneNote

Poznámkový blok OneNote umožňuje uspořádat úkoly v poznámkových blocích do oddílů, stránek i podstránek. Zahrnuje snadnou navigaci a vyhledávání. Poznámkové bloky je možné sdílet, a tak na projektech pracovat společně s ostatními členy týmu nebo k němu přistupovat přímo z aplikace Teams.

1.4 Výkaz práce, rozpočty a účetnictví

K založení společnosti také spadá neustálé řešení účetních výkazů, a to po celou dobu podnikání. Účetní standardy musí být v souladu s legislativou ČR.

Microsoft Excel

Tabulkový software Microsoft Excel lze využít pro vedení docházky zaměstnanců. Díky mnoha funkcím a možnosti tvoření dotazů lze v Excelu vytvářet přehledy pro výkazy práce a mzdy, ale také sledovat nákladovost projektů.

Microsoft Power BI

Interaktivní zobrazení dat a práce s nimi umí například Microsoft Power BI. Tento nástroj pro přehled a tvorbu grafů ale není ve standardní verzi balíčku Microsoft 365.

Altus Vario

Účetní systém je klíčovým bodem pro přehled mezd zaměstnanců, podrobného účetnictví, generování výkazů práce a faktur pro zákazníky. Příkladem může být Altus Vario nebo ABRA.

1.5 Uložiště dat

Pro různý typ shromažďovaných dat, k zajištění jejich dostupnosti a organizovanému přístupu, jsou navrženy nástroje s různými funkcemi. Ať už se jedná o ukládání běžných souborů nebo zdrojových kódů či data s běžným přístupem nebo zálohovaná data, je pro hledaný účel možné vybírat z celé řady aplikací.

Microsoft SharePoint

SharePoint je webová platforma určená k ukládání dat, jejich správě a sdílení s dalšími členy týmu. Data lze spravovat z jakéhokoli zařízení, přičemž je zajištěna bezpečnost a kontrola přístupu k datům. Webová stránka je přidružená vždy k týmu vytvořenému v Microsoft Teams v aplikaci Týmy, takže lze k datům přistupovat přímo z aplikace Teams. Díky seznamům a knihovnám SharePointu, Power Automate a Power Apps lze vytvářet fungující digitální prostředí s formuláři, pracovními postupy, automatickým schvalováním a vlastními aplikacemi pro každé zařízení.

Microsoft OneDrive

OneDrive je určen především jako osobní cloudové uložení, ve kterém je možné přistupovat ke svým souborům odkudkoliv. Při pravidelném zálohování se tak v případě ztráty zařízení předejde ztrátě dat. Jednotlivé soubory, ale i celé složky, lze sdílet napříč dalšími zaměstnanci a spolupracovat tak v reálném čase.

Azure DevOps

Azure DevOps je řešením pro verzování softwaru a správu verzí, testování, správu projektů pro agilní vývoj a vytváření sestav. Přehled práce je možno sledovat pomocí panelů Kanbanu, backlogů a dalších funkcí spadajících pod Azure Boards. Aktuální proces vývoje je udržován použitím gitu. Úplný přehled projektů, úkolů a produktů udržují grafy.

Azure Server

Platforma Azure od Microsoftu je služba pro hostování aplikací a dat v cloudu.

1.6 Prezentace společnosti

Webová stránka je určena pro prezentace a propagaci produktů a služeb, které společnost nabízí. Zobrazuje reference od zákazníků a poskytuje kontaktní informace zájemcům. Pro hosting webových serverů pro malé, střední i velké firmy se nabízí společnost Wedos.

2 Tvorba a integrace aplikací do prostředí Microsoft Teams

Microsoft Teams je rozšiřitelná platforma pro vytváření vlastních aplikací. Kromě komunikačního prostředí a spolupráce umožňuje integraci dalších aplikace tak, aby práce probíhala přehledně na jednom místě. Aplikace v MS Teams mohou být rozděleny do několika kategorií:

1. Vestavěné aplikace – jsou k dispozici pro všechny uživatele jako základní aplikace platformy Microsoft Teams. Lze sem přiřadit například aplikace Chat, Směny, Volání, Kalendář, Týmy a další.
2. Aplikace třetích stran – rozšiřují funkčnost MS Teams a umožňují uživatelům používat aplikace, které již znají a používají. Tyto aplikace mohou být zveřejněny ostatními organizacemi nebo získány z Microsoft AppSource či z jiných zdrojů.
3. Vlastní aplikace – umožňují týmům přizpůsobit si platformu Teams podle svých potřeb a vytvořit si vlastní procesy a funkce, které odpovídají specifickým potřebám týmu. Tyto aplikace lze vytvářet pomocí Microsoft Power Apps nebo SharePoint Framework a mohou být integrovány přímo do MS Teams.

2.1 Možnosti tvorby aplikací pro MS Teams

Způsob tvorby aplikací pro Microsoft Teams závisí na složitosti aplikace a jejích funkcí. Od zasílání oznámení kanálům nebo uživatelům, chatovacích botů nebo překladů jazyka až po integrované webové prostředí. Aplikace lze vytvářet pro jednotlivce, svůj tým, organizaci nebo pro všechny uživatele MS Teams.

1. Power Virtual Agents – je cloudová služba, která umožňuje vytvářet chatboty pro komunikaci s uživateli a poskytovat jim nápovědy nebo informace. Bota lze využít například pro zákaznický servis nebo pro plánování schůzek. Dokáže reagovat na klíčová slova, na základě kterých posílá konkrétní zprávy či spouští různé akce nebo toky v Power Automate. Nasazení chatbotů nevyžaduje jakoukoli zkušenost s kódováním.
2. Developer Portal – je nástroj pro vytváření aplikací pro Teams, které obsahují základní funkce, jako jsou karty, záložky, boti a rozšíření pro konverzace, vše bez nutnosti programování. Zároveň poskytuje centrální místo, kde vývojáři mohou vytvářet a testovat API pro vlastní naprogramované aplikace, spravovat přístupové klíče a autentizační tokeny.

3. Osobní aplikace – tvoří samostatnou aplikaci jako celek s pokročilejšími funkcemi. Použití Power Apps je ideální pro uživatele s pouze základními programátorskými dovednostmi, kteří chtějí vytvořit jednoduché aplikace. Pro pokročilejší aplikace se nabízí použití SharePoint Frameworku nebo vlastní webové aplikace, což už vyžaduje určitou úroveň znalostí vývoje webových aplikací a práce s API.

2.2 Způsob integrace aplikací do prostředí Microsoft Teams

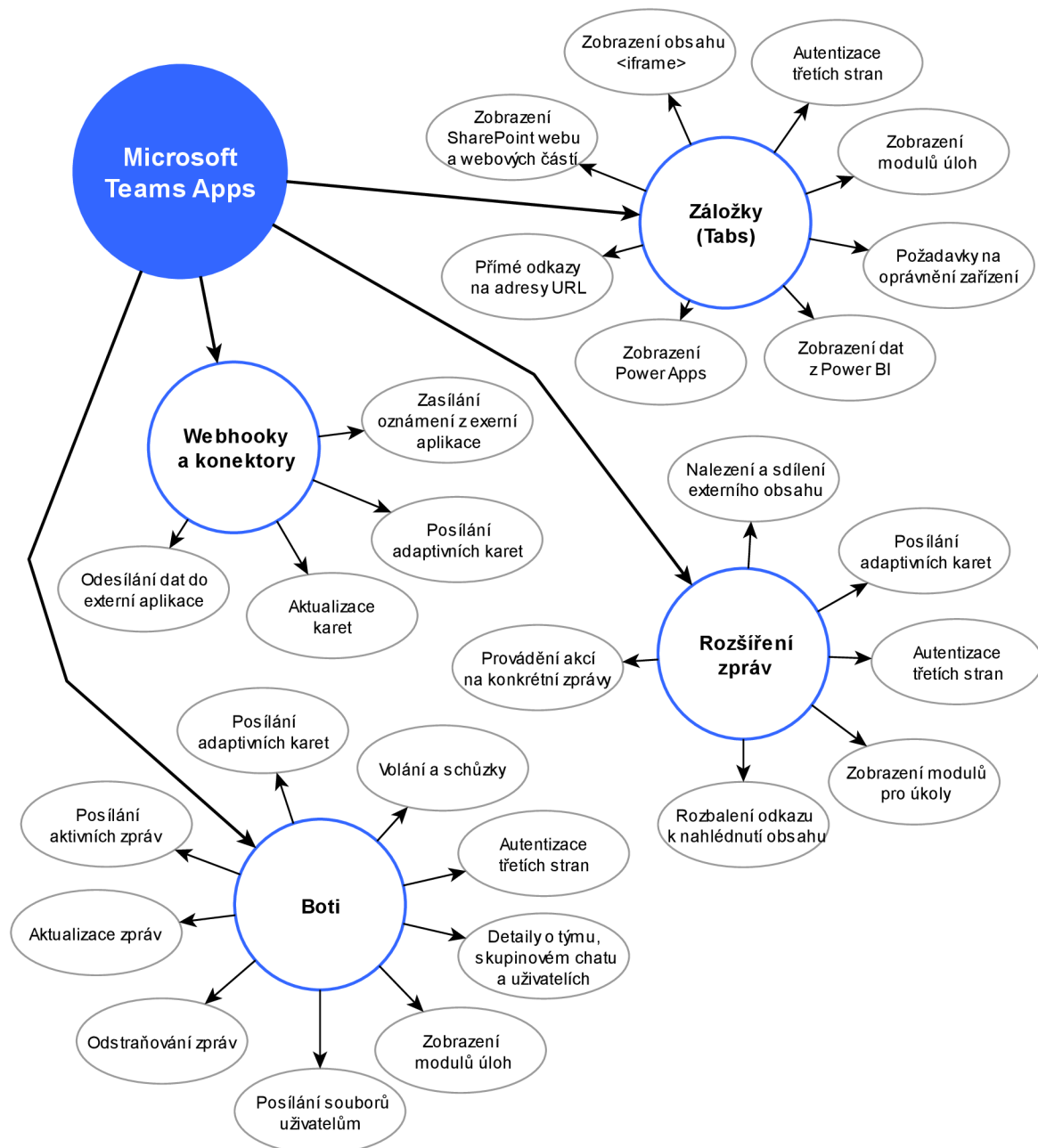
Integrace aplikací do prostředí Microsoft Teams může být provedena několika způsoby v závislosti na typu aplikace a jejích požadavcích na integraci.

1. Záložky (Tabs) – jsou stránky s webovým obsahem, které jsou zobrazeny přímo v MS Teams v rámci kanálu nebo chatu. Tuto funkci je možné využít například pro projektový management nebo sdílení dokumentů, seznamů či poznámek.
2. Karty (Cards) – jsou malé prvky uvnitř zpráv v Teams, které umožňují interakci s aplikacemi bez nutnosti opustit Teams. Mohou obsahovat třeba notifikace, drobné informace nebo interakci s dalšími aplikacemi, například s kalendářem.
3. Developer Portal – je nástroj pro tvorbu jednoduchých aplikací, případně je určen jako API prostředí pro webovou aplikaci hostovanou na SharePointu, v Azure nebo na jiném serveru. Obsahem balíčku pro aplikaci jsou ikony a manifest soubor s obsahem metadat, jako je název aplikace, popis, oprávnění či odkaz na ikony a samotnou aplikaci.
4. Power Platform – aplikace vytvořené pomocí Power Apps lze z tohoto nástroje automaticky nasadit do prostředí Teams. Po dokončení aplikace stačí nastavit cílovou platformu MS Teams, povolit aplikaci pro konkrétní uživatele nebo celou organizaci a publikovat. Aplikace lze zobrazit jako součást záložky v kanálu nebo jako samostatnou osobní aplikaci v levém navigačním panelu.
5. SharePoint App Catalog – je webová stránka pro správu a distribuci různých typů aplikací pro organizaci. Příkladem jsou webové části (webparts), „workflow“ aplikace, aplikace pro správu obsahu nebo vlastní aplikace. Publikované aplikace si uživatel může v MS Teams snadno najít a nainstalovat jako osobní aplikace do navigačního panelu.

Implementovat aplikaci do prostředí Teams se vyplatí v případě, že se jedná o denně využívanou aplikaci spouštěnou například při běžných obchodních procesech, aplikaci pro posílení komunikace, spolupráce a interakce nebo existující webovou aplikaci, Power App nebo web SharePoint rozšířený o SPFx. Pokud ale existuje již dobře navržená externí aplikace nebo se aplikace nevyužívá pravidelně, je vhodné zvážit jinou platformu.

2.3 Zmapování funkcí pro rozšíření vlastností Microsoft Teams

Microsoft Teams nabízí řadu možností, které umožňují vývojářům přidat různé funkce do svých aplikací a tím vytvářet výkonné prostředí, které uživatelům pomůže pracovat efektivněji a účinněji. [16]



Obrázek 1: Diagram funkcí pro rozšíření vlastností MS Teams

2.4 Osobní aplikace (Personal App)

Osobní aplikace postavené na platformě Microsoft Teams se skládají ze tří primárních komponent:

1. Klient Microsoft Teams – poskytuje body rozšíření a prvky uživatelského rozhraní, které aplikace použije k integraci s uživateli.
2. Balíček aplikace Teams – je nainstalován v MS Teams. Obsahuje malou ikonu, velkou ikonu a soubor JSON manifestu. Soubor manifestu obsahuje metadata aplikace (název, jméno vývojáře atd.), body rozšiřitelnosti aplikace (záložky, karty a rozšíření o zasílání zpráv) a odkazy na webové stránky (ID webových částí, URL adresa záložky).
3. Webové služby – poskytují rozhraní API a logiku aplikace. Platforma Teams není hostitelská služba, proto musí být webová služba hostovaná na vlastním serveru se zabezpečeným přístupem HTTPS přes internet, v Azure, na SharePointu nebo být součástí Power Apps.

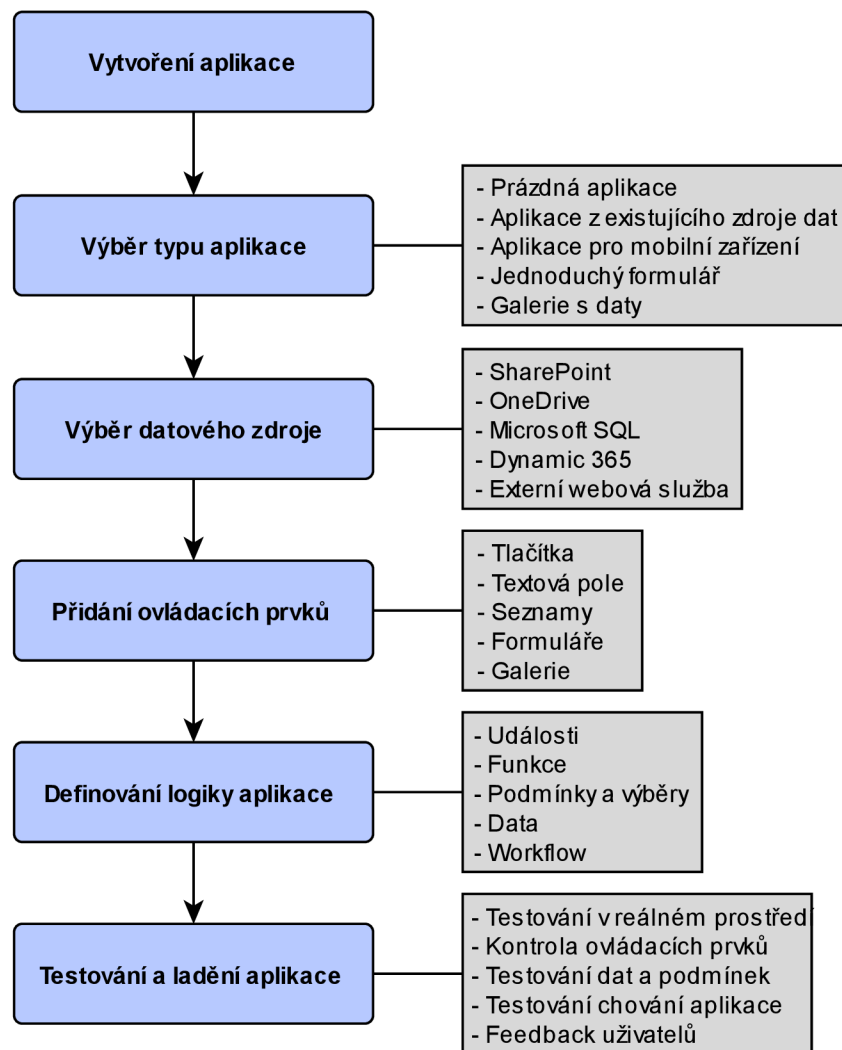
Existují tři hlavní způsoby, jak vytvořit a integrovat osobní aplikace s vlastní logikou do prostředí MS Teams – Power Apps, Sharepoint Framework Extensions (SPFx) nebo existující webová aplikace. Tyto způsoby tvorby a integrace aplikací jsem vyzkoušela, porovnála mezi sebou a sepsala limity jednotlivých řešení. Následující způsoby implementace a řešení problémů jsou tak pospány z pohledu autora a pro osvědčené postupy je třeba nahlédnout do oficiální dokumentace společnosti Microsoft. [20]

2.4.1 Power Apps

Power Apps je platforma pro tvorbu vlastních typů aplikací, včetně mobilních aplikací a webových aplikací. Umožňuje vytvářet základní aplikace bez nutnosti kódování, což je ideální pro uživatele s minimem programátorských dovedností. Power Apps nabízí možnost integrace s MS Teams, díky čemuž se aplikace zobrazuje přímo v kanále nebo chatu v Teams, případně ji lze zveřejnit jako samostatnou aplikaci v Teams. Tato integrace se provádí pomocí Power Apps komponenty pro Teams. Power Apps je velmi užitečná pro řešení konkrétních drobných úkolů a pro automatizaci určitých procesů. Některé funkce, jako třeba propojení se zdrojem dat z databáze, jsou prémiové a pro jejich používání je nutné zakoupit licenci. Příprava rozsáhlejších aplikací v Power Apps může být problematická z hlediska nedostatku volnosti při vytváření vlastní logiky programu, kdy s rostoucími požadavky roste i obtížnější nastavování vlastností funkcí a nutnost znalosti dané platformy.

2.4.1.1 Tvorba aplikací přes Power Apps

Power Apps nabízí intuitivní rozhraní, které umožňuje vytvořit aplikaci pomocí předem definovaných šablon nebo ji postavit od základů. Logiku aplikace lze rozšířit pomocí automatizovaných toků v Power Automate skládaných z funkčních bloků.



Obrázek 2: Use case postupu tvorby aplikace přes Power Apps

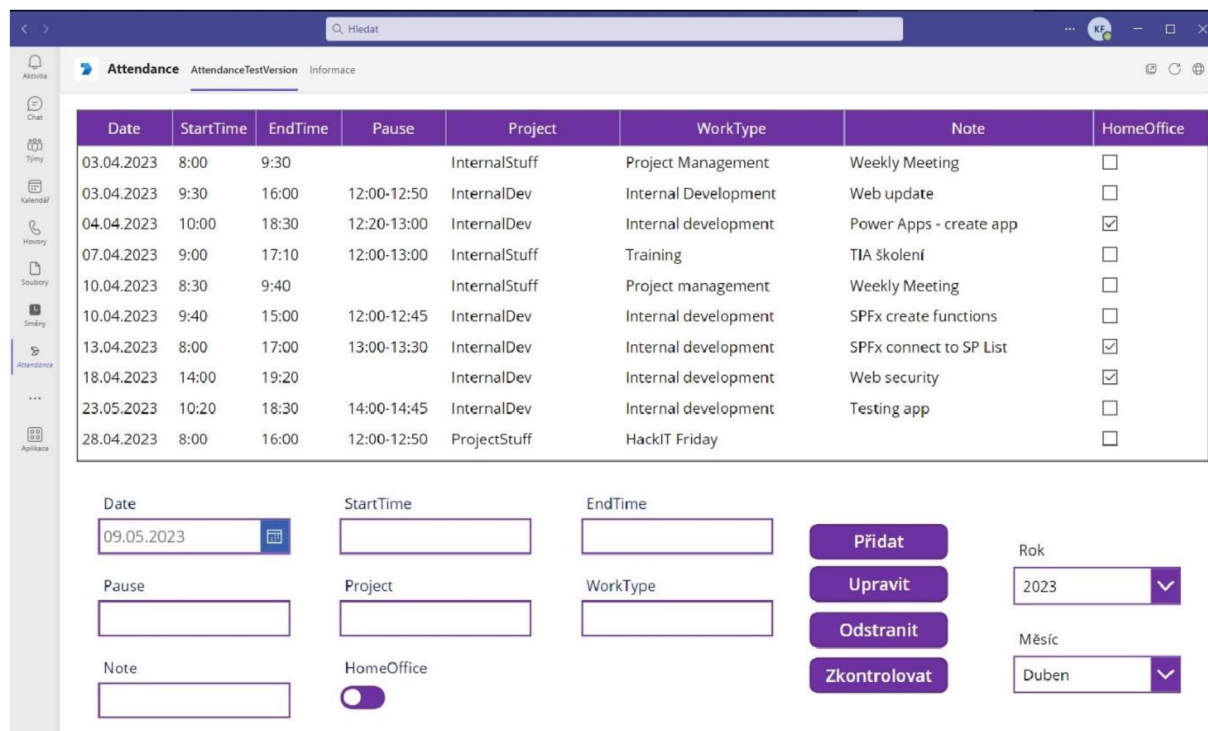
Tvorba aplikací v Power Apps se obvykle skládá z několika kroků. Nejprve je třeba definovat data a datové zdroje, které aplikace bude používat. Poté je nezbytné navrhnout uživatelské rozhraní aplikace, které zahrnuje výběr ovládacích prvků a jejich umístění na stránce. Dále je potřeba definovat logiku aplikace, která určuje, jak aplikace bude reagovat na uživatelské interakce a jak bude pracovat s daty. Poté se aplikace testuje a ladí, aby bylo zajištěno, že funguje správně a bez chyb.

1. Vytvoření aplikace – aplikace může být jako webová, desktopová nebo určena pro mobilní verzi.
2. Výběr typu aplikace – aplikaci lze vytvořit zcela od začátku, automaticky z návrhu v aplikaci Figma nebo pouze jako jednoduchý formulář či galerii s daty.
3. Výběr datového zdroje – je potřeba přidat propojení se službou pro ukládání dat a následně zvolit zdroj dat jako SharePoint, OneDrive, Microsoft SQL, Dynamic 365 nebo externí webová služba.
4. Přidání ovládacích prvků – uživatel ovládací prvky přetáhne z nabídky na plátno aplikace. Patří sem tlačítka, textová pole, seznamy, formuláře galerie a další.
5. Definování logiky aplikace – k vytvoření událostí k ovládacím prvkům, funkcím pro logiku nebo pro ukládání dat slouží předpřipravené nástroje jako události, funkce, podmínky a výběry, data a automatizované toky (workflow).
6. Testování a ladění aplikace – testování tlačítek a funkcí lze provádět přímo v prostředí Power Apps pomocí klávesy Alt. Část testování zahrnuje testování v reálném prostředí, kontrola ovládacích prvků, testování dat a připomínek, testování chování aplikace a feedback uživatelů.

Power Apps poskytuje nástroje a šablony, které usnadňují tyto kroky v procesu tvorby aplikací. Například disponuje nástrojem pro návrh uživatelského rozhraní pomocí přetahování a umístování ovládacích prvků „drag-and-drop“, stejně jako předdefinované šablony pro často používané typy aplikací, jako jsou např. formuláře a seznamy. Power Apps také umožňuje integraci s různými datovými zdroji, jako jsou např. SharePoint, SQL Server nebo jiné služby pro ukládání dat, což zjednodušuje přístup k datům a práci s nimi.

2.4.1.2 Integrace Power Apps aplikací do MS Teams

Přímo z Power Apps lze jedním tlačítkem automaticky publikovat aplikaci do MS Teams pro všechny členy organizace, případně nastavit omezení pro přístup k aplikaci. V samotném Teams lze aplikaci zobrazit pomocí záložky v některém z kanálů Týmu nebo přidat samostatně jako novou aplikaci v levém navigačním panelu.



Date	StartTime	EndTime	Pause	Project	WorkType	Note	HomeOffice
03.04.2023	8:00	9:30		InternalStuff	Project Management	Weekly Meeting	<input type="checkbox"/>
03.04.2023	9:30	16:00	12:00-12:50	InternalDev	Internal Development	Web update	<input type="checkbox"/>
04.04.2023	10:00	18:30	12:20-13:00	InternalDev	Internal development	Power Apps - create app	<input checked="" type="checkbox"/>
07.04.2023	9:00	17:10	12:00-13:00	InternalStuff	Training	TIA školení	<input type="checkbox"/>
10.04.2023	8:30	9:40		InternalStuff	Project management	Weekly Meeting	<input type="checkbox"/>
10.04.2023	9:40	15:00	12:00-12:45	InternalDev	Internal development	SPFx create functions	<input type="checkbox"/>
13.04.2023	8:00	17:00	13:00-13:30	InternalDev	Internal development	SPFx connect to SP List	<input checked="" type="checkbox"/>
18.04.2023	14:00	19:20		InternalDev	Internal development	Web security	<input checked="" type="checkbox"/>
23.05.2023	10:20	18:30	14:00-14:45	InternalDev	Internal development	Testing app	<input type="checkbox"/>
28.04.2023	8:00	16:00	12:00-12:50	ProjectStuff	HackIT Friday		<input type="checkbox"/>

Date	StartTime	EndTime	Přidat Upravit Odstranit Zkontrolovat	Rok
09.05.2023				2023
Pause	Project	WorkType		Měsíc
				Duben
Note	HomeOffice			
	<input type="checkbox"/>			

Obrázek 3: Náhled aplikace vytvořené v Power Apps a publikované v MS Teams

2.4.1.3 Výhody a nevýhody Power Apps

Power Apps umožňuje vytvořit jednoduché aplikace, jako formuláře nebo zobrazení seznamů během chvíle bez znalosti programování. Lze vytvářet i velmi pokročilé aplikace s náročnou logikou, ovšem to vyžaduje již pokročilé zkušenosti s vytvářením aplikací v Power Apps a základní znalosti programování. Ovšem v případě některých komplexnějších aplikací už lze narazit na limity nástroje Power Apps a je lepší zvolit jinou metodu tvorby aplikace. [3]

Tabulka 2: Výhody a nevýhody tvorby aplikací v Power Apps

Výhody	
Jednoduchá tvorba	Power Apps poskytuje způsob vytváření aplikací pomocí intuitivních nástrojů, proto i lidé bez hlubších programátorských znalostí mohou vytvářet jednoduché aplikace.
Rychlý vývoj	Vytváření jednoduchých aplikací v Power Apps je obvykle rychlejší než vývoj klasických aplikací, což může znamenat nižší náklady a rychlejší nasazení.
Minimální znalost Power Apps	Pro vývoj snadných aplikací v Power Apps není předchozí znalost vyžadována, jelikož si uživatel prostředí rychle osvojí. Pro složitější aplikace, ale už je vhodná znalost i dalších nástrojů Power Apps a zkušenost s nimi.
Snadná integrace s Microsoft 365 aplikacemi	Power Apps je dobře integrován s Microsoft 365 a dalšími produkty pomocí předpřipravených nástrojů, což umožňuje snadnou synchronizaci dat a procesů.
Připravené nástroje pro práci s Teams	Power Apps disponuje připravenými ovládacími prvky a funkcemi, které usnadňují tvorbu aplikací do MS Teams.
Připravené grafické prvky pro Teams	Power Apps má předpřipravené ovládací prvky ve stylu drag-and-drop rozhraní, které ladí s prvky v aplikacích Microsoft 365.
Bez potřeby registrace aplikace v Azure	Pro použití aplikace není potřeba aplikaci registrovat v Azure. Přímo z Power Apps je možné přidávat propojení s dalšími aplikacemi v rámci Microsoft 365. Může tak být ale omezena komunikace s dalšími aplikacemi třetích stran.
Vyřešené hostování	Vytvořená aplikace zůstává uložena v Power Apps.
Zabezpečení	Power Apps sám zajišťuje zabezpečení aplikací v MS Teams.
Správa přístupu	Přístup lze spravovat při publikování aplikace.
Snadná integrace do MS Teams	Snadná a rychlá integrace přes tlačítko Publikovat.
Snadné přidání aplikace	Aplikace se při publikování uživatelům zobrazí v Teams v seznamu aplikací.
Snadná údržba	Aktualizace a údržba aplikací v Power Apps je obvykle jednodušší a méně náročná než u tradičního vývoje.

Nevýhody	
Omezená přizpůsobivost aplikace	V případě potřeby složitějších nebo vysoce přizpůsobených funkcí mohou existovat omezení, stejně tak jako pro komunikaci s aplikacemi třetích stran.
Omezený design a přizpůsobení	Power Apps nabízí některé možnosti přizpůsobení designu a vzhledu aplikace, ale v případě složitějšího designu má svoje limity.
Malá komunitní podpora	Pro složitější práci s funkcemi a událostmi se hůře dohledají komunitní řešení, protože vývoj aplikací v Power Apps je dost specifický.
Závislost na platformě	Aplikace vytvořené v Power Apps jsou závislé na platformě Microsoft, což může omezit flexibilitu, pokud organizace používá jiné technologie.

Pro tvorbu jednoduchých a rychlých aplikací pro MS Teams je Power Apps ideálním nástrojem. Umožňuje vytvářet aplikace i uživatelům s minimem programátorských znalostí. Se zvyšujícími se nároky na aplikaci může být tvorba aplikace v Power Apps složitější, jelikož je potřeba dobře znát prostředí. Některé složitější scénáře sice lze v Power Apps vytvořit, ovšem v závěru to může být o dost náročnější než vytvoření funkcí u klasického programování. [9]

2.4.1.4 Limity low-code programování

Low-code je způsob návrhu a vývoje aplikace za pomoci grafických nástrojů a funkcí, které snižují požadavky na psaní kódu. Tento způsob tak umožňuje rychlou a jednoduchou tvorbu vlastních aplikací lidmi bez hlubších programátorských znalostí. Tyto nástroje mají ale omezené možnosti použití, jelikož uživatel je závislý pouze na tom, co daná platforma nabízí. V případě pokročilejších aplikací tak může snadno narazit na limity. [2]

Tabulka 3: Limity tvorby low-code aplikací, kterými jsou aplikace v Power Apps

Limity	
Automatické propojení se zdrojem dat	Nemůže automaticky vytvořit připojení k nové službě pro uchování dat. Když je vytvořen nový zdroj dat musí se ručně vytvořit nové připojení v aplikaci PowerApps, aby bylo možné zdroj dat použít.
Rozšířené zpracování dat	Má základních funkce pro zpracování dat, ale nepodporuje pokročilé techniky, jako například algoritmy pro třídění a vyhledávání dat.
Dynamické generování kódu	Nepodporuje dynamické generování kódu za běhu aplikace, což může být užitečné pro specifické úkoly, jako tvorbu dynamických dotazů do databáze.
Přístup k operačnímu systému zařízení	Neposkytuje možnost přímého přístupu k operačnímu systému zařízení, což znamená, že nelze použít funkce jako přístup k fotoaparátu nebo skeneru otisků prstů.
Vývoj aplikací s náročnými grafickými prvky	Nemá plnou podporu pro vývoj herních aplikací nebo aplikací, které vyžadují přístup k náročným grafickým prvkům.
Strojové učení a umělá inteligence	Umožňuje integraci s nástroji pro strojové učení a umělou inteligenci, ale nemá plnou podporu pro tvorbu vlastních modelů nebo algoritmů.
Integrace s externími systémy	Integruje s externími systémy pomocí různých konektorů a API, ale nemá podporu pro pokročilé techniky jako například pub/sub architekturu nebo komunikaci s RESTful API.
Pokročilé zabezpečení	Umožňuje řízení přístupu a ověřování identity, ale nemá podporu pro techniky jako například kryptografii nebo firewall.
Výkon a škálovatelnost	Může mít omezení výkonu a škálovatelnosti pro velké aplikace nebo aplikace s velkým objemem dat.
Asynchronní funkce a metody	Nemá plnou podporu pro pokročilé techniky asynchronního programování jako například asynchronní metody nebo funkce.

Práce s vlákny	Neumožňuje práci s vlákny pro řešení různých úkolů paralelně, jako například načítání velkého množství dat na pozadí a provádění jiných akcí v aplikaci – to zpomaluje běh aplikace.
Refactoring kódu	Nabízí základní refactoring jako například změna pořadí volání klíčových dat, ale neumožňuje rozdělení kódu do samostatných metod a tříd.
Plně objektově orientované programování	Nabízí některé funkce pro práci s objekty jako například vytváření a používání kolekcí, ale nepodporuje plně objektově orientované programování.
Rekurze	Neumí rekurzi, což může být problém při tvorbě složitějších aplikací. V takovém případě je třeba využít externího programování a připojení aplikace k API rozhraní.
Debugování kódu	Nástroj Monitor zobrazuje proud událostí z relace uživatele a informace o typu chyby a sleduje výkon aplikace. Nepodporuje pokročilé funkce jako například sledování hodnot proměnných a postup volání funkcí.

I přesto, že Power Apps nemusí plně podporovat všechny pokročilé funkce a programovací techniky důležité pro vývoj sofistikovaných a komplexních aplikací, stále poskytuje dostatek nástrojů a funkcí pro tvorbu uživatelsky přívětivých aplikací pro firemní procesy a řízení dat bez nutnosti znalosti programování.

2.4.2 SharePoint Framework (SPFx)

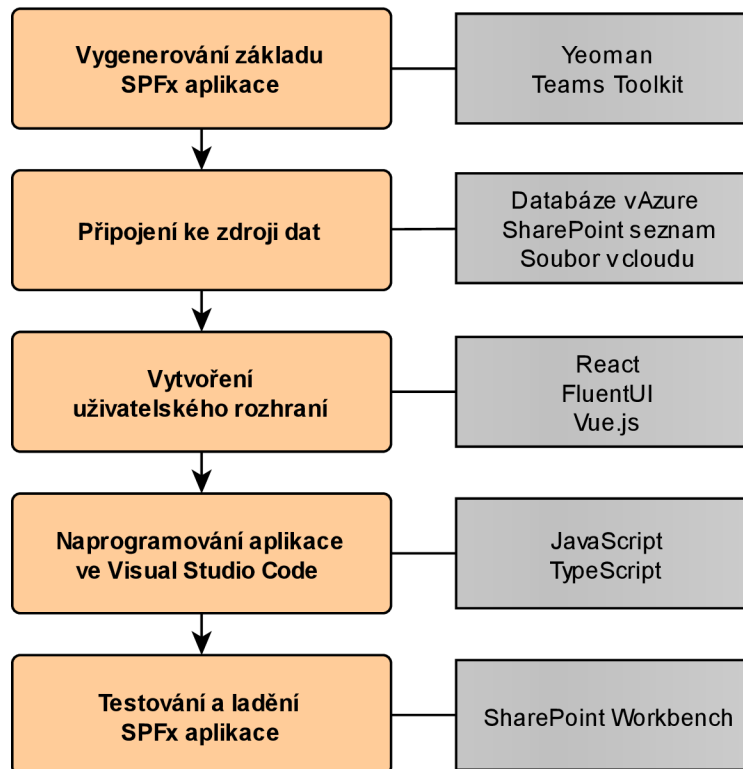
SharePoint Framework (SPFx) je moderní vývojový framework pro vývoj webových a mobilních aplikací pro SharePoint Online a SharePoint Server 2019. SPFx umožňuje vývojářům vytvářet prvky webových stránek, webové části, rozšíření pro SharePoint a další aplikace pomocí webových technologií, nástrojů a frameworků, jako jsou například TypeScript, React, Node.js, Gulp nebo FluentUI. Aplikace jsou snadno přenositelné mezi různými prostředím a jsou navrženy tak, aby je bylo možné snadno spravovat a aktualizovat.

Dalším využitím SPFx je možnost rozšířit stávající funkcionalitu SharePointu, přidat nové prvky do stránek nebo upravit stávající webové části o specifické vlastnosti a funkce, které jsou pro danou organizaci klíčové.

SPFx může být také využito pro integraci s dalšími službami a aplikacemi, jako je například Teams nebo Power Apps. Vývojáři mohou vytvářet vlastní aplikace a prvky, které jsou navrženy tak, aby fungovaly společně s těmito službami a maximalizovaly tak efektivitu a produktivitu uživatelů.

2.4.2.1 Tvorba aplikací pomocí SPFx

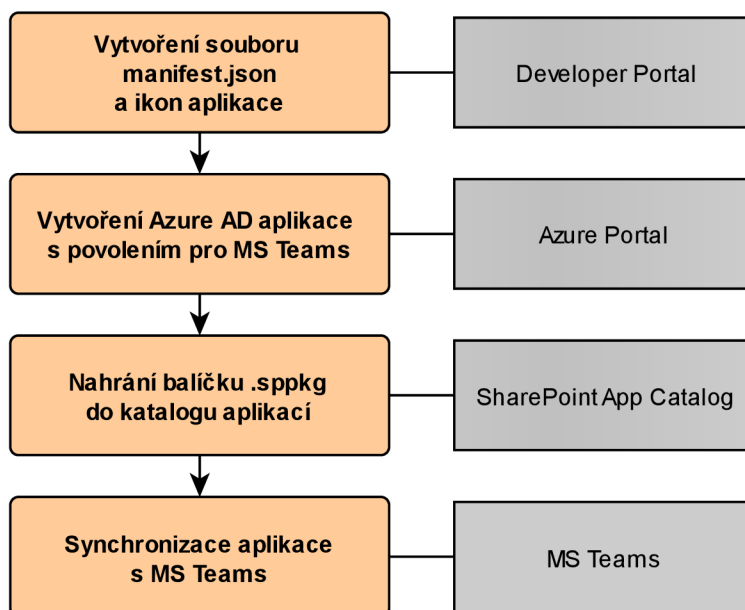
SPFx umožňuje vytvářet webové části, které lze snadno integrovat do SharePointu, ale také do MS Teams. Tato integrace se provádí pomocí SharePoint komponenty pro Teams. Základní struktura aplikace se vygeneruje pomocí open source generátoru aplikací Yeoman za použití „@microsoft/sharepoint“. Návrh a vývoj aplikace probíhá ve vývojovém prostředí Visual Studio Code v jazyce JavaScript nebo TypeScript. Pro tvorbu uživatelského rozhraní se využívá React a pro designové prvky lze použít FluentUI, který je navržený tak, aby prvky vytvářely sjednocené prostředí s MS Teams. VS Code je možné rozšířit o SharePoint Framework Property Pane Assistant, který umožňuje vývojářům snadno vytvářet uživatelská rozhraní pro nastavení vlastností webových částí. Pro propojení SPFx řešení se SharePoint seznamy se používá třída SPHttpClient, pro přístup k dalším základním aplikacím Microsoft 365 pak třída MS Graph API. Pro komunikaci s Azure API je to AadHttpClient. Testovat aplikaci lze lokálně pomocí SharePoint Workbench nebo přímo v MS Teams.



Obrázek 4: Use case postupu tvorby aplikace pomocí SPFx

2.4.2.2 Způsob integrace SPFx aplikací do MS Teams

Pro hotovou aplikaci je nutné připravit JSON soubor manifestu, aby bylo možné aplikaci integrovat do MS Teams. Manifest obsahuje metadata o aplikaci, informace o webových částí a data k integraci do prostředí MS Teams. Lze ho vytvořit ručně nebo předpřipravit v nástroji Developer Portal v MS Teams a následně stáhnout a vložit do aplikace i s ikonami. Pokud aplikace pro komunikaci s dalšími aplikacemi Microsoft 365 nebo s Azure API využívá MS Graph API nebo AadHttpClient, je potřeba v Azure Portal registrovat novou aplikaci, která je propojená s SPFx řešením a tuto komunikaci povoluje. Z aplikace je potřeba vytvořit balíček a ten vložit na SharePoint do App Catalog, odkud se integruje do MS Teams. V samotném Teams si už uživatelé mohou aplikaci přiřadit a nainstalovat.



Obrázek 5: Use case postupu integrace SPFx aplikace do MS Teams

2.4.2.3 Výhody a nevýhody SPFx

Použití SPFx pro tvorbu aplikací do Microsoft Teams má několik nevýhod, zejména co se týče omezených funkcí MS Teams a potřeby znalosti SharePointu a vývoje v něm. Přesto však SPFx poskytuje výhody, jako jsou nástroje a knihovny pro tvorbu aplikací, snadná integrace, široká podpora technologií, komunity a otevřenost kódu. Umožňuje vývojářům vytvářet aplikace pro Teams a SharePoint na jedné platformě, což usnadňuje správu a aktualizaci aplikací. Je užitečným nástrojem pro vývojáře, kteří chtějí vytvářet i složitější aplikace a funkce pro Teams nebo SharePoint web.

Tabulka 4: Výhody a nevýhody použití SPFx pro tvorbu aplikací

Výhody	
Snadná integrace s Microsoft 365 aplikacemi	SPFx dokáže dobře komunikovat s dalšími aplikacemi v rámci Microsoft 365 pomocí připravených dostupných knihoven, frameworků a tříd jako SPHttpClient a MS Graph API.
Připravené nástroje pro práci s Teams	SPFx poskytuje připravené nástroje a knihovny, které usnadňují tvorbu aplikací do MS Teams.
Minimálně omezená přizpůsobivost aplikace	Vývojář má vlastní kontrolu nad kódem a může si funkce upravovat podle potřeby. Podporuje různé jazyky jako TypeScript, JavaScript nebo vlastní tvorbu CSS. Pro některé složitější scénáře nebo integrace s třetími stranami ale může SPFx dosáhnout svých limitů.
Připravené grafické prvky pro Teams	Díky frontend frameworku FluentUI, ve kterém je připravený design Microsoft 365 aplikací, dokážou prvky a samotná aplikace dobře ladit s prostředím Teams.
Vlastní design a přizpůsobení	SPFx poskytuje podporu pro širokou řadu moderních technologií, včetně React, Angular, Vue.js nebo FluentUI a vzhled aplikace je tak čistě na vývojáři.
Vysoká komunitní podpora	SPFx je open source projekt, což znamená, že vývojáři mohou přispět kódem a vylepšovat existující nástroje a knihovny, což také znamená, že vývojáři mohou najít spoustu zdrojů a komunitních řešení.
Vyřešené hostování	Hotová aplikace se pouze nahraje do SharePoint App Catalog.
Zabezpečení	SPFx poskytuje možnosti pro zabezpečení aplikací v MS Teams, včetně správy přístupu, šifrování a ochrany dat.
Správa přístupu	Omezení přístupu k aplikaci pouze pro určité uživatele nebo týmy lze spravovat z Azure Portal.
Snadné přidání aplikace	Aplikace je hostována v SharePointu a uživatelé si tak aplikaci mohou snadno najít v seznamu aplikací vytvořených pro organizaci a používat bez nutnosti dalších nastavení.
Sdílení kódu	Vývojáři mohou sdílet kód mezi aplikacemi vytvořenými pro SharePoint a MS Teams, což znamená menší úsilí při vývoji a aktualizaci aplikací.
Široké spektrum možností	Použití SPFx umožňuje vývojářům vytvářet různé typy aplikací do MS Teams, včetně záložek, botů a rozšíření pro kanály.

Nevýhody	
Složitější tvorba	I když SPFX poskytuje nástroje a knihovny pro tvorbu aplikací, vytváření pomocí SPFx může být pro vývojáře začátečníky stále složité a náročné.
Pomalejší vývoj	Vývoj SPFx aplikací je složitější a vyžaduje programátorské dovednosti, proto je i časově náročnější, což znamená také vyšší náklady a delší čas pro nasazení.
Nutná znalost SPFx	Pro vývoj aplikací v SPFx je potřeba mít znalosti s tímto frameworkem, což může být problém pro vývojáře bez předchozích zkušeností s SPFx.
Registrace aplikace v Azure	Pro řízení oprávnění SPFx aplikace je aplikace spojená s registrací aplikace „ <i>SharePoint Online Client Extensibility Web Application Principal</i> “, která řídí přístup k rozhraní API jako Microsoft Graph API, vlastnímu API hostovanému v Azure nebo jiným cloudovým službám Microsoftu.
Odborná integrace do MS Teams	Vyžaduje odborné znalosti a práci se SharePointem pro nastavení integrace, což může zvyšovat náklady na tvorbu aplikace.
Závislost na platformě	Aplikace vytvořené v SPFx jsou závislé na platformě Microsoft, což omezuje jejich použití při přechodu na jiné technologie.
Náročnější údržba	Pomocí SPFx je možné vytvářet aplikace pro MS Teams a SharePoint na jedné platformě, což umožňuje snadnou správu a aktualizaci aplikací. Ovšem může mít omezenou podporu pro funkce, což může vést k problémům s kompatibilitou při aktualizaci MS Teams. Aktualizovanou verzi je potřeba vždy znovu nahrát do SharePoint App Catalog.

2.4.3 Existující webová aplikace

Pokud organizace má vlastní již existující webovou aplikaci, je možné tuto aplikaci integrovat do Microsoft Teams pomocí vlastní záložky (Tabs), karty (Cards) nebo jako osobní aplikaci (Personal App) do levého navigačního panelu.

- Vytvoření vlastní záložky umožňuje zobrazovat webovou aplikaci přímo v kanále nebo skupinovém chatu. Stačí přidat záložku v kanálu nebo chatu, vybrat typ „Web“ a vložit URL webové stránky.
- Vytvoření vlastní karty umožňuje uživatelům interagovat s webovou aplikací přímo v rámci konverzace. Adaptivní kartu lze vytvořit pomocí toků v Power Automate, které přímo nabízí blok pro vytvoření adaptivní karty. Sem vložit kód pro zobrazení webové stránky nebo vytvořit karty s obsahem vygenerovaným v JSON formátu, například formuláře, ankety, denní přehledy atd.
- Vytvoření vlastní osobní aplikace je vhodné v případě, že se jedná o rozsáhlejší webovou aplikaci jako celek s vlastní pokročilou logikou a je vhodné ji mít oddělenou samostatně jako další aplikaci v MS Teams. Pro implementaci se využívá nástroj Developer Portal.

2.4.3.1 Tvorba webové aplikace

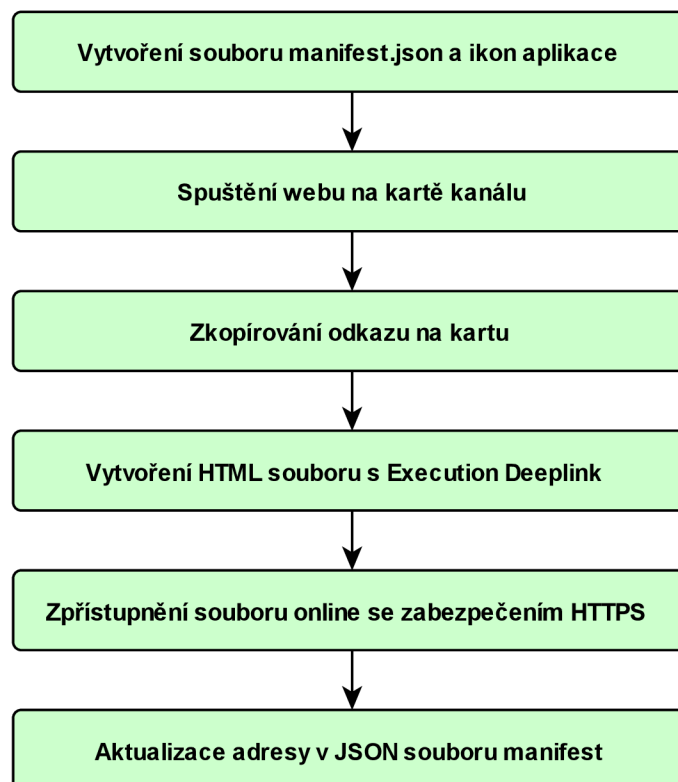
Webová aplikace pro integraci s MS Teams by měla být navržena tak, aby uživatelům poskytovala snadný přístup ke svým funkcím a datům přímo v rámci Teams.

- Podporuje SSO (Single Sign-On), což umožní uživatelům přihlásit se jednou do MS Teams a poté automaticky získat přístup k aplikaci bez dalšího přihlašování.
- Má responsivní design, aby byla dobře použitelná na různých zařízeních, včetně mobilních telefonů a tabletů.
- Má silné zabezpečení a jsou přijímána vhodná opatření pro ochranu dat a přístupů.
- Poskytuje API nebo jiné rozhraní, které umožní komunikaci s Microsoft Teams. To zahrnuje způsob, jakým bude aplikace získávat a zobrazovat data v Teams.
- Měla by poskytnout možnost konfigurovat a přizpůsobit chování v rámci Teams, jako možnosti nastavení administrátorů pro správu aplikace pro celou organizaci.

- Má k dispozici dokumentaci pro uživatele i administrátory, která popisuje, jak používat či případně konfigurovat aplikaci.
- Před integrací by měla být webová aplikace prověřena na testovacím prostředí Teams, pro ujištění, že vše funguje správně.
- Je zvolený způsob, jakým budou uživatelé interagovat s aplikací, ať už se jedná o vytváření záložek, botů nebo jiných prvků pro přístup uživatelů k funkcím aplikace.

2.4.3.2 Způsob integrace existující webové aplikace do MS Teams

Firemní webovou stránku nebo webovou aplikaci hostovanou již na jiném serveru je možné zpřístupnit z MS Teams přímo z levého panelu aplikací. To lze jednoduše přes vložení odkazu na stránku do manifest souboru vytvořeném v nástroji Developer Portal v MS Teams. Ovšem pokud webová aplikace obsahuje funkci zabezpečení, která blokuje jejich načtení do prvku iFrame, odkaz na stránku nebude fungovat. Lze to ale vyřešit jiným způsobem popsaném níže.



Obrázek 6: Use case postupu integrace existující webové aplikace do MS Teams

1. Vytvoření souboru manifest.json a ikon aplikace

Microsoft Teams obsahuje aplikaci s názvem Developer Portal, dříve App Studio. V tomto nástroji je možné vytvořit prostředí pro přidání aplikace nebo propojení existující hostované webové aplikace. Při vytvoření je nutné zadat metadata aplikace, vygenerovat ID aplikace, nastavit oprávnění pro běh aplikace, vložit ikony, a především přidat osobní karty s ID entitou a URL adresou na webovou stránku, kterou je potřeba propojit.

Nyní stačí kartu uložit, aplikaci dokončit a testovat nebo publikovat. Pokud se ale stránka nezobrazuje, obsahuje funkci zabezpečení, která blokuje její načtení do prvku iFrame a je tak nutné přejít k dalším krokům níže.

Osobní karty a aplikace v Teams jsou implementovány pomocí prvku iFrame. To je HTML tag, který se používá pro vložení jiných dokumentů nebo stránek do aktuální webové stránky. Jelikož mnoho webů má funkci zabezpečení, která blokuje jejich načtení do prvku iFrame, a osobní karty Teams jsou implementovány pomocí prvku iFrame, pak zobrazení této webové stránky nebude fungovat.

Pro zjištění, zda problém nezobrazování stránky je opravdu z důvodu prvku iFrame, stačí vytvořit HTML soubor s tagem iFrame, vložit do něj URL adresu webu a soubor otevřít v prohlížeči.

```
<html>
  <body>
    <iframe src="URL_ADRESA_WEBU" width=800 height=600></iframe>
  </body>
</html>
```

Zdrojový kód 1: HTML soubor s tagem iFrame

Obsahem hlavičky HTTP, která je odeslána webovým serverem hostujícím web, který se nezobrazuje, je „X-Frame-Options“. S hodnotou „SAMEORIGIN“ dává prohlížeči pokyn, aby zablokoval zobrazení webu v prvcích iFrame hostovaných mimo weby se stejným názvem domény jako web, který je třeba zobrazit. iFrame by tedy fungoval pouze v případě, že by byl web hostován na libovolném webu *.microsoft.com.

2. Spuštění webu na kartě kanálu

Karty v kanálech nevyužívají prvky iFrame na rozdíl od osobních karet. Načítají weby stejně jako prohlížeč, takže web načtený aplikací Web na kartě v kanálu Teams nepodléhá omezením „X-Frame-Options“.

Stačí tak v libovolném kanálu v Teams vytvořit novou záložku (Tab) typu Web a tam vložit URL adresu webu k zobrazení v Teams.

3. Zkopírování odkazu na kartu

Nyní je v kartě spuštěna chráněná webová stránka. Je potřeba si zkopírovat odkaz na tuto kartu s webem a ten si uložit.

4. Vytvoření HTML souboru s „Execution Deeplink“

Následně je potřeba vytvořit HTML soubor s javascriptovým kódem s pokynem k provedení přímého odkazu na kartu, na které je spuštěný web. HTML kód načte sadu SDK klienta Teams a provede instrukci pro přechod na kartu. SDK obsahuje různé API pro přístup k datům a funkcím MS Teams, jako jsou například kanály, konverzace, soubory a kalendáře. Také poskytuje řadu nástrojů pro vývoj a testování aplikací včetně nástrojů pro vytváření uživatelských rozhraní, ladění a ověřování aplikací. Používá se pro vytvoření různých aplikací a integrací pro MS Teams, jako jsou například chatboti, aplikace pro správu projektů a integrace se službami třetích stran.

```
<html>
  <head>
    <title>Teams Deep Link Example</title>
    <script src= 'https://statics.teams.cdn.office.net/sdk/v1.6.0/
      js/MicrosoftTeams.min.js'></script>
  </head>
  <body>
    <script>
      microsoftTeams.initialize();
      microsoftTeams.executeDeepLink('ADRESA_Z_KARTY');
    </script>
  </body>
</html>
```

Zdrojový kód 2: HTML soubor s „Execution Deeplink“ na kartu v kanálu Teams

5. Zpřístupnění souboru online se zabezpečením HTTPS

Soubor je potřeba zpřístupnit z Teams. Aplikace volá tento HTML soubor a k tomu potřebuje způsob, jak obsloužit požadavek HTTPS. Soubor je možné hostovat v Azure Server, na cloudovém serveru nebo dokonce na místním webovém serveru, který je schopen hostovat jedinou statickou HTML stránku prostřednictvím protokolu HTTPS.

6. Aktualizace adresy v JSON souboru manifest

Následně pouze stačí se vrátit do aplikace vytvořené v Developer Portal a v manifest.json souboru změnit URL adresu na web s nově hostovaným HTML souborem, aplikaci uložit a publikovat. Ta se po instalaci zobrazí v MS Teams v levém panelu aplikací. Pokaždé, když se klikne na ikonu aplikace, klient Teams spustí kód hostované vytvořené HTML stránky a přejde na kartu v kanálu, kde se web načítá. [24]

2.4.3.3 Výhody a nevýhody existující webové aplikace

Pokud organizace má už svou vlastní webovou aplikaci nebo využívá webovou aplikaci třetích stran, může ji implementovat přímo do Teams jako samostatnou aplikaci nebo záložku v kanálu Týmů. Tím se vyhne programování aplikace. Lze vytvořit i novou samostatnou aplikaci s vlastními funkcemi a bez omezení, na kterých je možné narazit při tvorbě aplikace v SPFx nebo Power Apps. Ovšem je potřeba mít tyto aplikace již hostované, což zvyšuje náklady stejně tak, jako náklady na tvorbu webové aplikace, která vyžaduje odborné zkušenosti vývojáře. Externí webová aplikace obvykle postrádá API pro komunikaci s dalšími aplikacemi Microsoft 365. Dále je potřeba se do této aplikace neustále přihlašovat, případně mít nastavený automatický přístup k aplikaci bez dalšího přihlašování. Aplikace by měla být dobře zabezpečená. Například nelze implementovat aplikaci bez zabezpečené komunikace HTTPS.

Tabulka 5: Výhody a nevýhody použití existující webové aplikace

Výhody	
Neomezená přizpůsobivost aplikace	Při tvorbě vlastní webové aplikace není vývojář nijak omezen funkcemi a může vytvářet i složitější scénáře a aplikaci si zcela přizpůsobit vlastním potřebám.
Vlastní design a přizpůsobení	Při vývoji frontend části u své webové aplikace není vývojář nijak omezen vzhledem prostředím a grafických prvků a může si tak vzhled a rozložení aplikace přizpůsobit podle potřeby.
Vysoká komunitní podpora	Vývojář si pro tvorbu aplikace může zvolit jaký jazyk chce a pro vývoj klasické webové aplikace dohledá dost komunitních řešení.
Bez registrace aplikace v Azure	Existující webová aplikace nepotřebuje pro integraci do Teams být registrována v Azure.
Snadná integrace do MS Teams	Pokud je aplikace již hostovaná a přístupná se zabezpečením HTTPS a neblokuje prvek iFrame, je její integrace do MS Teams přes Developer Portal jednoduchá. V opačném případě je potřeba učinit nastavení, které vyžadují odbornější znalost.
Snadné přidání aplikace	Aplikace si po přidání do Developer Portal mohou uživatelé snadno přidat mezi své aplikace v Teams.
Nezávislost na platformě	Běh vlastní webové aplikace není závislý pouze na platformě Microsoft, ale aplikace může fungovat i samostatně.
Snadná údržba	Jelikož externí webová aplikace nijak není propojena s Teams, není potřeba řešit údržbu fungování aplikace přímo v Teams.
Různorodost aplikací	Je možné vytvořit vlastní aplikaci nebo využít už existující aplikaci třetích stran pro integraci do Teams.

Nevýhody	
Náročná tvorba	Vytvoření vlastní webové aplikace je určeno pro zkušené vývojáře s hlubšími programátorskými znalostmi. V případě již existující vlastní webové aplikace nebo aplikace třetích stran je tak příprava aplikace téměř bez práce.
Pomalý vývoj	Vývoj klasických webových aplikací je náročnější s potřebou odborných programátorských znalostí, proto je i časově náročný a nákladný s delší dobou pro nasazení.
Nutná znalost vývoje webových aplikací	Pro vývoj vlastních aplikací je potřeba mít odborné znalosti a zkušenosti s vývojem webových aplikací. V případě použití již existující aplikace nejsou odborné znalosti potřeba.
Bez integrace s Microsoft 365 aplikacemi	Klasická webová aplikace, ať už vlastní nebo existující, obvykle nepodporují nástroje pro komunikaci s aplikacemi v rámci Microsoft 365, především kvůli zabezpečení. Je ale možné vytvořit API rozhraní hostované v Azure pro přístup k aplikacím Microsoft 365.
Žádné nástroje pro práci s Teams	Vývoj klasické aplikace nezahrnuje nástroje pro práci s aplikací v rámci Teams.
Žádné grafické prvky pro Teams	Klasická aplikace nemusí být navržena tak, aby ladila s prostředím MS Teams, ale stále může použít frameworky, kterými se k designu Teams přiblíží.
Vlastní hostování	Pro vlastní aplikaci je potřeba si vyřešit vlastní hostování, které bývá i placené. Pro aplikaci třetích stran není potřeba hostování řešit, pokud aplikace neblokuje zobrazení v iFrame. V takovém případě je nutné vytvořit API, které je nutné hostovat.
Vyřešit vlastní zabezpečení	Je nutné mít dobře nastavené zabezpečení aplikace. Stejně tak musí být řádně zabezpečena i integrace, aby se minimalizovala rizika spojená s datovými ztrátami a zneužitím.
Omezená správa přístupu	Nelze spravovat přístup k aplikaci dalšími členy organizace. Výjimkou je vytvoření aplikace registrované a hostované v Azure, která je prostředníkem existující webové aplikace a umožňuje správu přístupu k aplikaci.

2.4.4 Srovnání způsobů implementace aplikací do MS Teams

Existuje několik způsobů, jak tvořit a integrovat interní aplikace do prostředí MS Teams.

1. Power Apps

- Rychlá tvorba „low-code“ aplikací pomocí „drag-and-drop“ rozhraní s nízkou náročností vývoje bez znalosti programování.
- V případě složitějších aplikací je potřeba základní znalost programování a dobrá znalost a zkušenost s prostředím Power Apps.
- Tvorba desktopových, mobilních i webových aplikací.
- Omezená přizpůsobivost aplikace – je potřeba aplikaci sestavit z již připravených komponent a datových modelů. Je omezená kontrola nad logikou aplikace.
- Integrace s Teams se provádí pomocí Power Apps komponenty.

2. SharePoint Framework (SPFx)

- Střední náročnost vývoje, vyžaduje znalost SharePointu a programování pomocí TypeScriptu nebo JavaScriptu s využitím dalších frameworků.
- Podpora vlastních datových modelů, UI prvků a logiky.
- Poskytuje vývojářům větší kontrolu nad kódem, a tak i vysokou přizpůsobivost.
- Tvorba celé aplikace, webových částí nebo pouze rozšíření do SharePointu webu.
- Integrace do Teams přes SharePoint App Catalog.

3. Vlastní webová aplikace

- Vhodné v případě, že má organizace již vlastní webovou aplikaci.
- Vyžaduje vysokou náročnost vývoje, dobrou znalost programování a webových technologií a zkušenosti s tvorbou webových aplikací.
- Nemá podporu pro Teams specifické funkce, jako jsou „custom“ záložky a chatboti ani designové prvky pro sjednocení s prostředím MS Teams.
- Výhodou integrování existujících webových aplikací do Teams je, že umožňuje vývojářům využívat již existující kód a funkce aplikace.
- Integrace do MS Teams přes odkaz pomocí záložky nebo karty.
- Integrace jako osobní aplikace pomocí iFrame nebo Teams Web View.

Každá z možností má své výhody a nevýhody a vyžaduje určitou úroveň technických dovedností. Použití Power Apps je ideální pro uživatele bez programátorských dovedností, kteří chtějí vytvořit jednoduché aplikace. Použití SharePoint Framework a vlastní webové aplikace vyžaduje určitou úroveň znalostí vývoje webových aplikací a práce s API.

Tabulka 6: Srovnání vlastností pro tvorbu aplikací do MS Teams v Power Apps, pomocí SPFx a vložením existující webové aplikace

Funkce	Power Apps	SPFx aplikace	Existující/vlastní webová aplikace
Potřebné znalosti programování	Nízké – vytváření pomocí drag-and-drop rozhraní, low-code znalosti	Střední – znalost programování, funkce pro práci s SP a Graph API	Vysoké – pokročilá znalost programování a vývoje aplikací
Doba vývoje	Rychlá – pro snadné aplikace.	Střední – pro pokročilejší aplikaci	Dlouhá – pro vývoj celé aplikace
Potřebné zkušenosti s daným způsobem tvorby aplikací	Žádné pro základní aplikace, střední pro pokročilé aplikace a funkce	Střední – dobrá znalost prostředí SharePointu a tvorby SPFx aplikací	Vysoké – zkušenosti s vývojem webových aplikací
Integrace s Microsoft 365 aplikacemi	Dostupná – připravené nástroje pro komunikaci	Dostupná – třídy, knihovny a frameworky	Minimální – omezená integrace s dalšími aplikacemi
Nástroje pro práci s Teams	Rozšířené – připravené na vytváření custom aplikací	Rozšířené – připravené na tvorbu aplikací, webových částí nebo rozšíření pro SharePoint	Žádné – bez Teams specifických funkcí
Přizpůsobivost aplikace	Omezená – lze vytvářet z již připravených komponent	Vysoká – vlastní kontrola nad kódem, omezení na aplikaci zanedbatelné	Vysoká – tvorba aplikace bez omezení
Grafické prvky pro Teams	Všechny – připravené pro tvorbu aplikací do MS Teams	Omezené – některé ve FluentUI pro sjednocení prvků s MS Teams	Žádné – vlastní design

Design a přizpůsobení	Omezené – lze pouze vybírat z předpřipravených šablon a ovládacích prvků	Střední – vlastní výběr vzhledu aplikace, ale nutno dodržet strukturu	Vysoké – vzhled a struktura aplikace bez omezení
Komunitní podpora	Nízká – vývoj je dost specifický	Střední – je možné dohledat návody a komunitní řešení	Vysoká – pro klasický vývoj webových aplikací je spousta návodů
Registrace aplikace v Azure	Ne – vytváří si vlastní propojení	Ano – pro propojení s dalšími aplikacemi	Ne – integrace bez registrace
Hostování aplikace	V Power Apps	V SharePoint App Catalog	Vlastní server na hosting
Zabezpečení	Vyřešené	Vyřešené, ale při komunikaci s vlastním API nutno zajistit	Nutno zajistit
Správa přístupu	Možná při publikování	Možná přes Azure registraci aplikací	Nelze, případně to lze zajistit pomocí aplikace hostované v Azure
Integrace s Teams	Snadná – zařizuje Power Apps	Střední – potřeba odborné znalosti SharePointu	Snadná – vložení odkazem přes Developer Portal
Přidání aplikace	Snadné	Snadné	Snadné
Závislost na platformě	Závislá na prostředí Teams	Závislá na prostředí Teams nebo SharePoint	Nezávislá na platformě
Údržba a aktualizace	Snadná	Náročnější obzvláště při aktualizacích MS Teams	Bez údržby v případě existující aplikace

3 Docházkové systémy a stávající procesy pro vedení docházky v MS Teams

Platforma Microsoft Teams nedisponuje dostatečně pokročilým docházkovým systémem, který by byl součástí koupeného balíčku. Nemá přímo zabudované nebo oficiálně řešené vedení docházky přímo v MS Teams. To znamená, že většina malých a středních podniků využívá buď tabulkový Excel, jeden soubor pro jednoho zaměstnance, který je vytvořený na cloudovém uložišti a musí se manuálně propojit s novými projekty, které jsou vytvářeny v rámci Microsoft Teams, nebo si firmy platí externí software bez automatického spárování s projekty, pokud se nejedná o pokročilý systém s API. Nutností je platit si licenci a vzniká tak další aplikace, do které je nutné se přihlašovat a která není integrována do MS Teams.

3.1 Existující systémy pro vedení docházky

K účinné správě pracovního času zaměstnanců organizace existuje na trhu široká škála systémů. Jejich funkce od možnosti sledování přítomnosti, evidence odpracovaných hodin po plánování směn a generování reportů poskytují efektivní přehled pracovní produktivity. Mohou integrovat s dalšími systémy, jako jsou nástroje pro fakturaci a platby, a tím usnadnit administrativní procesy. Pro výběru vhodného systému je potřeba zohlednit vlastní specifické potřeby, velikost týmu a požadavky na sledování produktivity a nákladů.

MS Shifts

Aplikace Směny přímo v MS Teams umožňuje záznam docházky, ale pouze jako zaregistrování příchodu, odchodu a zahájení přestávky a její ukončení a následný export pracovního výkazu ve formátu PDF. Každému uživateli se v této aplikaci Směny dají přiřadit předem naplánované směny, podle kterých se pak řídí jeho týdenní rozvrh. V těchto směnách je možné ručně zapsat, na jakých projektech pracuje. Také zde může žádat o volno kvůli lékaři nebo dovolené. Podle jeho výkazu práce a směn je pak orientačně možné získat přibližný přehled o tom, kolik reálných hodin na daném projektu mohl strávit. Ovšem směny a výkaz práce nejsou nijak spárované, tudíž není možné se tímto způsobem řídit.

TimeTracker

TimeTracker umožňuje členům týmu zaznamenávat docházku přes SharePoint Modern WebPart. Aplikace je integrována do MS Teams a je k dispozici pro mobilní zařízení, tablety, desktopové aplikace i weby. Pravidla a zásady pro přestávky a pracovní dobu jsou zajištěny dodržováním předpisů a bezpečnosti práce. Všechna data jsou následně exportována do Excel souboru pro analýzu a výplatní listinu. Data z docházky jsou uložena do seznamů na Sharepointu. Aplikace je zdarma, ale nemá plnou servisní podporu. [1]

Timeneye

Timeneye je docházkový systém integrovatelný přímo do MS Teams. Funguje na principu dashboardu, který zobrazuje přehled o docházce na každý týden. Do dashboardu je možné ručně přidávat tabulky se záznamem času na daném projektu, časy různě měnit a upravovat. Zobrazuje grafy přehledu a statistik odpracovaných hodin na různých projektech i sledování výkazu na konkrétním projektu. Aplikace je placená – 6 dolarů měsíčně za uživatele. [25]

AttendanceBot

AttendanceBot se integruje přímo do pracovního prostředí MS Teams, kde si zaměstnanci mohou zaznamenávat docházku pomocí chatování s botem. Pro pokročilejší nastavení a přehledy je uživatel odkázán na webovou stránku. Oprávněný uživatel má přístup k hodinám zaměstnance přes řídicí panel pro analýzu pracovní doby. Základní verze je za 4 dolary ročně na uživatele. [4]

Docházka Lokace

Je mini aplikace se záznamem příchodu do zaměstnání a odchodu z něj. Hodí se především pro firmy, které se více zdržují v terénu než v kanceláři. Aplikace má jednotnou cenu, a to 15 tisíc Kč bez DPH. [5]

Výkaz práce

Je software pro evidenci práce a docházky, který je funkcemi blízko k pokročilejšímu systému. Jako externí software umožňuje evidenci docházky i na odloučených pracovištích, zobrazuje práci na zakázkách, probíhajících činnostech i výdaje. Jeho obsahem jsou i podklady pro mzdy a fakturaci. Zaznamenává knihu jízd a veškeré přehledy a vyhodnocení zobrazuje v Excel souboru nebo PDF. Cena se odvíjí od počtu aktivních uživatelů, pro jednoho uživatele je v plné verzi cena 80 Kč za měsíc. [22]

GIRITON

Docházka GIRITON je velmi šikovný pokročilý externí software pro záznam docházky. Má záznam docházky na klik tlačítka příchod a odchod, pauzou nebo tlačítkem HomeOffice a přehled odpracovaných hodin za daný den. Dále graf s odpracovanou dobou a přehledem pro dovolené, lékaře a HomeOffice nebo případné chyby v docházce. Zobrazuje vlastní plán směn, součty hodin na daných směnách a plánování dovolených. Je zde výkaz práce na projektech, zobrazení přesčasů nebo zobrazení problémových záznamů. Docházka se dá zaznamenávat z mobilu, pískaček, desktopové aplikace i webového prohlížeče. Také funguje automatické vložení GPS docházky na základě vstupu nebo opuštění předem vyznačených oblastí. Docházka GIRITON zahrnuje stravenky, bonusy a odměny. Oprávněným uživatelům zobrazuje docházku ostatních zaměstnanců a umožňuje importy a exporty docházky a její tisk. Docházka GIRITON je velmi mocný nástroj s velkým množstvím funkcí a pěknými přehledy a grafy. Cena se odvíjí od počtu uživatelů, základní cena pro jednoho uživatele je 495 Kč za měsíčně, za každého dalšího uživatele je to plus 35 Kč za měsíc. [6]

Odoo Timesheets

Timesheets od Odoo je určen zaměstnancům organizace k evidenci pracovního času, což usnadňuje proces sledování produktivity a fakturace klientům za odvedenou práci. Klíčovou vlastností Odoo Timesheets je možnost sledovat a zaznamenávat svůj čas prostřednictvím mobilních zařízení, což zvyšuje flexibilitu pracovních postupů. Dále podporuje schvalovací procesy, tedy všechny záznamy o odpracovaném čase procházejí schvalovacím postupem před samotnou fakturací. Díky tomu si organizace udržuje přehled o nákladech. Odoo zahrnuje celou sadu dalších aplikací pro řízení firmy, což z něj dělá sjednocený systém. Standardní balíček těchto aplikací je v hodnotě cca 300 Kč za uživatele na měsíc při platbě na rok předem, rozšířená varianta pak za cca 430 Kč za uživatele na měsíc při platbě na rok předem. Za využití pouze jedné aplikace od Odoo s neomezeně uživateli je částka 0 Kč.

3.1.1 Srovnání existujících systémů pro vedení docházky

Na platformě Microsoft Teams jsou vyvíjeny jinými firmami i další aplikace fungující jako docházka, ovšem jejich použití nemusí být dostatečné, jelikož z nich nelze následně získávat přehledy dat pro další zpracování. Dále neumožňují zadávání takových dat, která jsou pro danou organizaci potřebná ani nejsou propojeny se seznamem projektů. Externí docházkové systémy většinou obsahují všechny funkce, ovšem již nejsou součástí Teams a mají vyšší náklady. Na základě toho vzniká potřeba vytvoření vlastního docházkového systému na míru přesně podle kritérií firmy.

Tabulka 7: Srovnání vybraných existujících systémů pro vedení docházky

	Pokročilé funkce	Spárování s projekty	Zadávání odkudkoliv	Přizpůsobené exporty	Součást Teams	Placená
MS Shifts	Ne	Ne	Ano	Ne	Ano	Ne
TimeTracker	Ne	Ne	Ano	Ano	Ano	Ne
Timeneye	Ano	Ne	Ano	Ano	Ano	Ano
AttendanceBot	Ano	Ne	Ano	Ne	Ano	Ano
Lokace	Ne	Ne	Ano	Ne	Ne	Ano
GIRITON	Ano	Ne	Ano	Ano	Ne	Ano
Výkaz práce	Ano	Ne	Ano	Ano	Ne	Ano
Odo	Ano	Ne	Ano	Ano	Ne	Ne

3.2 Stávající procesy pro vedení docházky v Microsoft 365

Microsoft 365 neobsahuje pokročilý software pro vedení docházky, proto aktuální softwarové řešení týkající se docházky obvykle funguje v tabulkovém softwaru Microsoft Excel. Příkladem může být ručně vytvořený nový Excel soubor ze šablony pro každého zaměstnance na každý nový rok. V tom si zaměstnanec sám vede svou docházku. U každého souboru Excel je potřeba ručně nastavit oprávnění, aby k tomuto souboru měl přístup pouze ten zaměstnanec, kterému je docházka určena. Každý list značí jeden měsíc. V tomto souboru v daném měsíci si pak zaměstnanec v pracovních dnech dopisuje řádky, kde ve sloupcích definuje:

- Datum
- Začátek práce a její konec
- Začátek přestávky a její konec
- Projekt – vybírá ze seznamu projektů definovaných na prvním listu Excelu (tento seznam si přebírá seznamy projektů ze SharePoint webu nebo jiného Excel souboru, tudíž se tento seznam aktualizuje)
- Typ práce – opět výběr z předpřipraveného seznamu
- Poznámky

Záznamy docházky se ručně kontrolují podle pravidel a zásad pro pracovní dobu. Pro přestávky je hrazeno minimálně 30 minut denně a je nutné mít zadané výjezdy o sobotách, nedělích a svátcích a práce přes noc, které jsou hrazené příplatkem ve mzdě.

	A	C	D	E	F	G	H	I	J	K
1	Day	Start	End	Brake Start	Brake End	Brake Tot	Total a Day	Project	Work Type	Notes
2	02.01.2023	8:00	8:30			0:00	0:30	InternalDe	Internal development	Power Automate - New Team
3	02.01.2023	8:30	9:45			0:00	1:15	InternalStu	Project management	Weekly Meeting
4	02.01.2023	9:45	18:40	12:30	13:00	0:30	8:25	InternalDe	Internal development	Power Automate - Create New Team
5	04.01.2023	12:00	17:55			0:00	5:55	InternalDe	Internal development	SP Lists, Power Automate
6	06.01.2023	12:25	18:20			0:00	5:55	InternalDe	Internal development	PA New Team, Shifts
7	08.01.2023	12:50	14:10			0:00	1:20	InternalDe	Internal development	(HO) SPOMS SP List Templates
8	09.01.2023	8:35	9:35			0:00	1:00	InternalStu	Project management	Weekly Meeting
9	09.01.2023	9:50	15:05	12:15	13:00	0:45	4:30	InternalDe	Internal development	New Team, Shifts
10	09.01.2023	15:05	17:00			0:00	1:55	InternalDe	Internal development	Meeting PA New Team + Shifts
11	09.01.2023	17:00	18:45			0:00	1:45	InternalDe	Internal development	PA New Team
12	10.01.2023	1:00	1:30			0:00	0:30	InternalDe	Internal development	(HO) Auto create shifts
13	12.01.2023	15:30	23:59	17:20	19:20	2:00	6:29	InternalDe	Internal development	(HO) PA Shifts
14	13.01.2023	0:10	5:05	2:00	2:30	0:30	4:25	InternalDe	Internal development	(HO) PA Shifts
15	13.01.2023	13:50	17:50			0:00	4:00	nternalDev	Internal development	PA Shifts
16	15.01.2023	16:15	17:30			0:00	1:15	InternalDe	Internal development	Set New Team
17	16.01.2023	8:35	9:55			0:00	1:20	InternalStu	Project management	Weekly Meeting
18	16.01.2023	9:55	17:20	14:15	14:45	0:30	6:55	InternalDe	Internal development	PA Shifts
19	23.01.2023	8:50	9:40			0:00	0:50	InternalStu	Project management	Weekly Meeting
20	23.01.2023	9:40	20:00	12:00	12:50	0:50	9:30	InternalDe	Internal development	Web, Power Automate, Shifts
21	29.01.2023	15:10	22:15	18:15	20:00	1:45	5:20	InternalDe	Internal development	(HO) Merge Excel, Web, Shifts
22	30.01.2023	11:20	17:55	12:00	12:50	0:50	5:45	InternalDe	Internal development	Merge Excel, Web, Shifts
23	30.01.2023	22:10	23:00			0:00	0:50	InternalDe	Internal development	(HO) PA Shifts (deleate option)
24	31.01.2023	11:05	23:05	16:10	20:50	4:40	7:20	InternalDe	Internal development	(HO) PA Shifts

Obrázek 7: Přehled fungování docházky v souboru MS Excel

3.3 Důvody vytvoření pokročilého docházkového systému

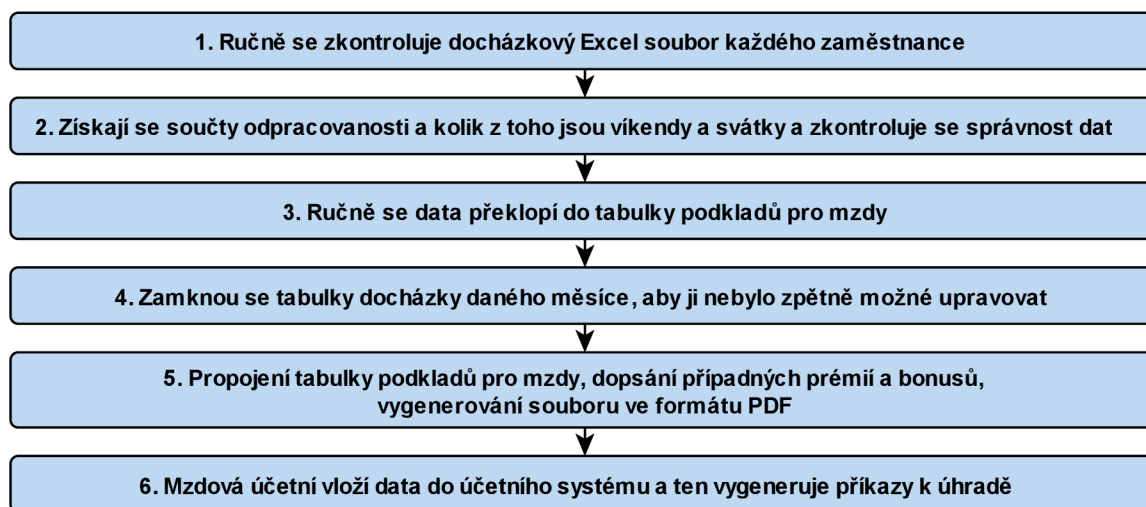
Hlavním problémem, který vedl k myšlence předělání docházkového systému, je její přehled a spárování s projekty. Pro získání přehledu práce na projektu pro zákazníka je potřeba ručně připravit dokument s přehledem dnů práce na projektu a celkovém počtu hodin strávených nad daným projektem. Dále je potřeba ručně z docházek sepsat hodiny věnované konkrétním milníkům projektu atd. Vytvoření těchto výpisů ručně je zdlouhavé. Stejný problém vzniká při kontrole docházky z předchozího měsíce na začátku každého měsíce, kdy je potřeba provést kontroly výkazů u každého zaměstnance.

Dále je to důvod zatím neexistujícího pokročilého softwaru pro vedení docházky na platformě MS Teams ani celého balíčku Microsoft 365 tak, aby se k docházce dalo přistupovat jednoduše a přehledně z jednoho místa, organizace měla jednoduchý přístup ke svým datům a aby se nemusela platit licence externího docházkového softwaru.

Už existující softwarová řešení docházky implementovaná do MS Teams umí pouze jednoduché zaznamenání docházky a nemají žádné další funkce na generování následujících přehledů. Tyto přehledy jsou popsány jako příklady fungování generování mezd, výkazů práce a dalších funkcí v případě vedení docházky v souboru MS Excel.

3.3.1 Získání přehledu pro mzdy

Každý zaměstnanec má svůj Excel soubor, ve kterém si vede docházku. Tento soubor je rozdělen na měsíce, kdy každý list v souboru je jeden měsíc. Na každý rok je pro zaměstnance jeden Excel soubor. Ten se na nový rok vždy musí pro každého zaměstnance vygenerovat ze šablony a nastavit oprávnění, kdo si tento soubor může zobrazit. V novém měsíci se vždy musí zkontrolovat docházky všech zaměstnanců z předchozího měsíce a vygenerovat přehledy.



Obrázek 8: Postup pro získání přehledu pro mzdy z docházky

Tato práce zabere měsíčně celé 2–3 dny člověku pro správu dokumentů. Vhodné řešení by byla automatická kontrola na klik a vygenerování přehledu do Excel souboru nebo PDF.

3.3.2 Vygenerování výkazu práce pro zákazníka

Zákazník platí buď za celé dílo nebo za hodiny strávené danou zakázkou.

- Dílo má pevně danou cenovku s pevně daným zadáním. Zákazník se nezajímá o počet hodin strávených na takovém projektu. Příkladem je návrh řešení integrace robota.
- Aktivity jako servis, podpora, dozory apod. se platí v závislosti na stráveném čase programátora dané aktivity – den či týden výjezdu. Člověk se účtuje v hodinách. Nastaví se cena programátora na hodinu. Obvykle toto funguje při výjezdech do zahraničí, které zahrnují náhrady za stravné, cestu, ubytování atd.

U aktivit po ukončení výjezdu, prací, smluvně dohodnutém milníku, po změně kalendářního měsíce či předání zakázky zákazníkovi je potřeba vygenerovat servisní protokol nebo výkaz prací, který zákazník podepíše a na základě kterého je možné poslat fakturu a následně peníze.

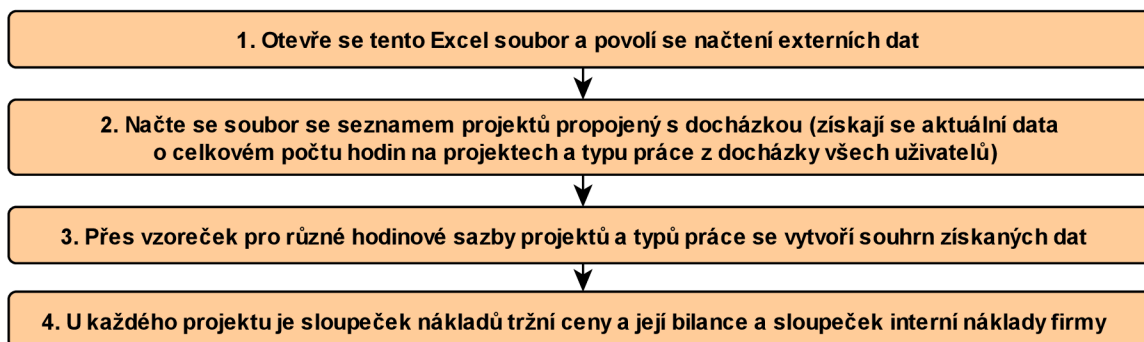


Obrázek 9: Postup pro vygenerování výkazu práce pro zákazníka

Tato akce pro vygenerování jednoho výkazu práce zabere 4–5 hodin v závislosti na období pro generování výkazu či náročnosti projektu. Pokud by tyto kroky byly generovány automaticky na klik, ušetřilo by to několik hodin práce.

3.3.3 Interní sledování nákladů projektů

V Excel tabulce pro interní sledování nákladovosti projektů jsou zapsaní jednotliví zaměstnanci a souhrn všech projektů z Excelu pro seznam projektů (s tímto souborem je propojena i docházka).

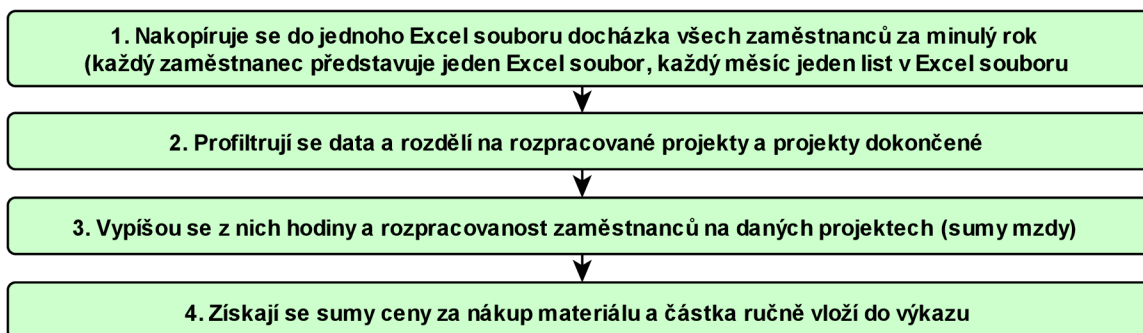


Obrázek 10: Postup pro získání interní sledovanosti nákladů projektu

Vhodné by bylo propojení s účetním systémem.

3.3.4 Vygenerování účetní rozpracovanosti

Jsou dva dokumenty účetní uzávěrky – účetní rozvaha a výkaz zisku a ztráty. Účetní rozvaha je nedokončená výroba a polotovary, kam spadají veškeré vynaložené náklady, jako nákup materiálu a mzdy. Projekt začíná objednávkou zákazníka a uhrazením zálohové faktury jako souhlasu s provedením díla. Zálohy uhradí mzdy a nakupený materiál. Po dokončení zakázky a posláním konečné faktury a jejím uhrazení zákazníkem se projekt promítá do tržeb a zakázka je dokončena – do té doby je rozpracovanou zakázkou. Rozpracovaností je např. vývoj softwaru, kdy veškeré mzdové náklady, výdaje za nákupy HW, komponent a materiálu za rok se sečtou, v případě nákupů bez DPH.

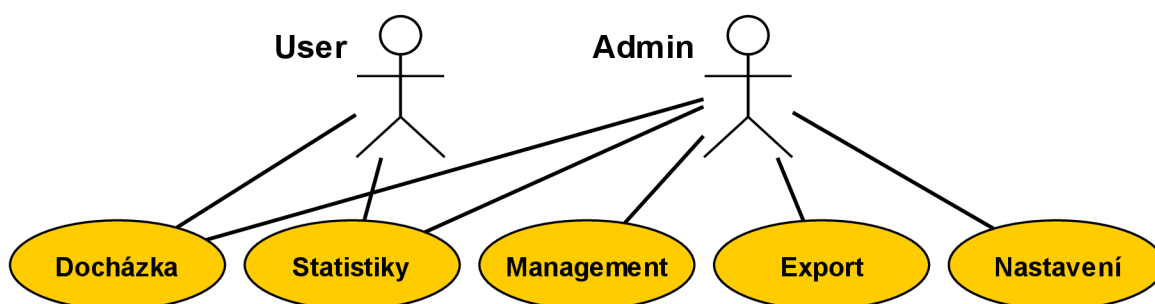


Obrázek 11: Postup pro vygenerování účetní rozpracovanosti

Toto trvá až celkem 3 dny práce. Docílení získání dat k rozpracovanému projektu na klik je akcí na 10 minut. Existují docházkové systémy, které toto umožňují, ale problém je s propojením dat, se získáním další externí aplikace nespádající pod Microsoft 365, s další aplikací jako službou a její licenci a platbou za ní a další aplikací, kam by se zaměstnanci museli přihlašovat.

4 Funkční požadavky pro docházkový systém

Cílem je vytvořit pokročilý docházkový systém implementovaný do prostředí Microsoft Teams. To znamená, že kromě klasického záznamu docházky umí i další funkce. Umožňuje zaměstnancům zápis docházky přiřazené k projektům, zobrazování přehledů, správu docházky a generování dokumentů. Admin dědí práva uživatele, a navíc má přístup k záložkám Management, Export a Nastavení.



Obrázek 12: Diagram přehledu oprávnění pro uživatele a správce

Požadavkem je snížit přepínání kontextu a nutnost platit licenci dalšího softwaru pro vedení docházky, tudíž mít aplikaci přehledně na platformě MS Teams. Je důležité rozlišovat mezi sekcí pro uživatele a správce. Podle legislativních požadavků může uživatel vidět pouze svou docházku. Dále musí být dodržena přestávka během pracovní doba a to po 4 hodinách, nejpozději po 6 hodinách. U zápisu docházky je potřeba rozlišovat, zda se jedná o práci z domova. Dále je nutné kontrolovat, zda je směna v sobotu, v neděli, o svátcích nebo během noční, jelikož za takové směny jsou příplatky.

V docházce musí probíhat kontrola, tedy validace záznamu a doplnění všech potřebných informací, jako je typ práce a projekt. Ty se nezdvávají ručně, ale jsou propojeny s existujícím seznamem, aby uživatel vybíral z již připraveného seznamu a mohlo dojít tak k párování s projekty. Docházka by také měla mít možnost rychlého záznamu přes tlačítka v případě, že se člověk nachází mimo kancelář.

Správce má možnost nahlížet do docházek všech uživatelů a provádět případné úpravy. Také může zamykat docházky v případě, že je ukončen měsíc, aby v nich nebylo možné staré záznamy uživatelem měnit. Správce získává přehled o statistikách k danému měsíci uživatele.

Hlavním důvodem, proč vytvořit vlastní docházkový systém je rozšíření o tlačítka na automatické generování dokumentů z docházky pro výpisy pro mzdy, výkazy práce na daném projektu pro zákazníka, účetní rozpracovanost i přehled nákladovosti projektů. To ušetří až několik dnů práce měsíčně lidem, kteří tyto věci aktuálně spravují ručně.

Tabulka 8: Funkční požadavky na docházku

FUNKCE PRO ZAMĚSTNANCE:	Záložka	Priorita
Přidat směnu do docházky ručně	Docházka	1
Upravit vybranou směnu v docházce	Docházka	2
Odstranit směnu z docházky	Docházka	3
Zvolit rok a měsíc zobrazované docházky	Docházka	4
Zahájit a ukončit směnu a spravovat přestávky přes tlačítka a tím je automaticky přidat do docházky	Docházka	5
Označit směnu jako HomeOffice	Docházka	6
Zadat více směn k jednomu datu a více přestávek k jedné směně	Docházka	7
Zobrazit přehled počtu odpracovaných hodin za daný měsíc	Statistiky	8
FUNKCE PRO SPRÁVCE:		
Zobrazit docházku všech zaměstnanců	Management	9
Přidat směnu do docházky zaměstnanců	Management	10
Upravit docházku zaměstnanců	Management	11
Odstranit docházku zaměstnanců	Management	12
Uzamknout zkontrolované docházky	Management	13
Upravit naložení s přesčasem	Nastavení	14
Vygenerovat přehled pro mzdy	Export	15
Vygenerovat výkaz práce na projektu podle kritérií	Export	16
Vygenerovat nákladovost projektu	Export	17
Vygenerovat účetní rozpracovanost	Export	18

5 Návrh aplikace docházkového systému

Aplikace sestává ze šesti záložek – samotné docházky (Docházka), přehledů za měsíc (Statistiky), docházky všech uživatelů (Management), výpisů (Export), nastavení (Nastavení) a informací o aplikaci (O aplikaci). Informace o aplikaci jsou jako záložka generovány automaticky a obsahují pouze základní popis aplikace, jméno vývojáře a vložené odkazy.

- První záložka zobrazuje samotnou docházku právě přihlášeného uživatele na platformě Teams. Umožňuje zobrazení měsíců, zápis i úpravy docházky a její záznam na klik.
- Druhá záložka je zobrazením přehledů k docházce za měsíc.
- Třetí záložka je dostupná pouze pro správce a je určena k přehledu a úpravám docházky všech uživatelů a následně uzamknutí docházky, aby nedocházelo k dalším úpravám.
- Čtvrtá záložka je také pouze pro správce a je určena pro generování výpisů z docházky ve formátu PDF nebo do CSV či případně do tabulkového souboru Excel.
- Pátá záložka je určena pro správce k práci se zdrojem dat a k základnímu nastavení.

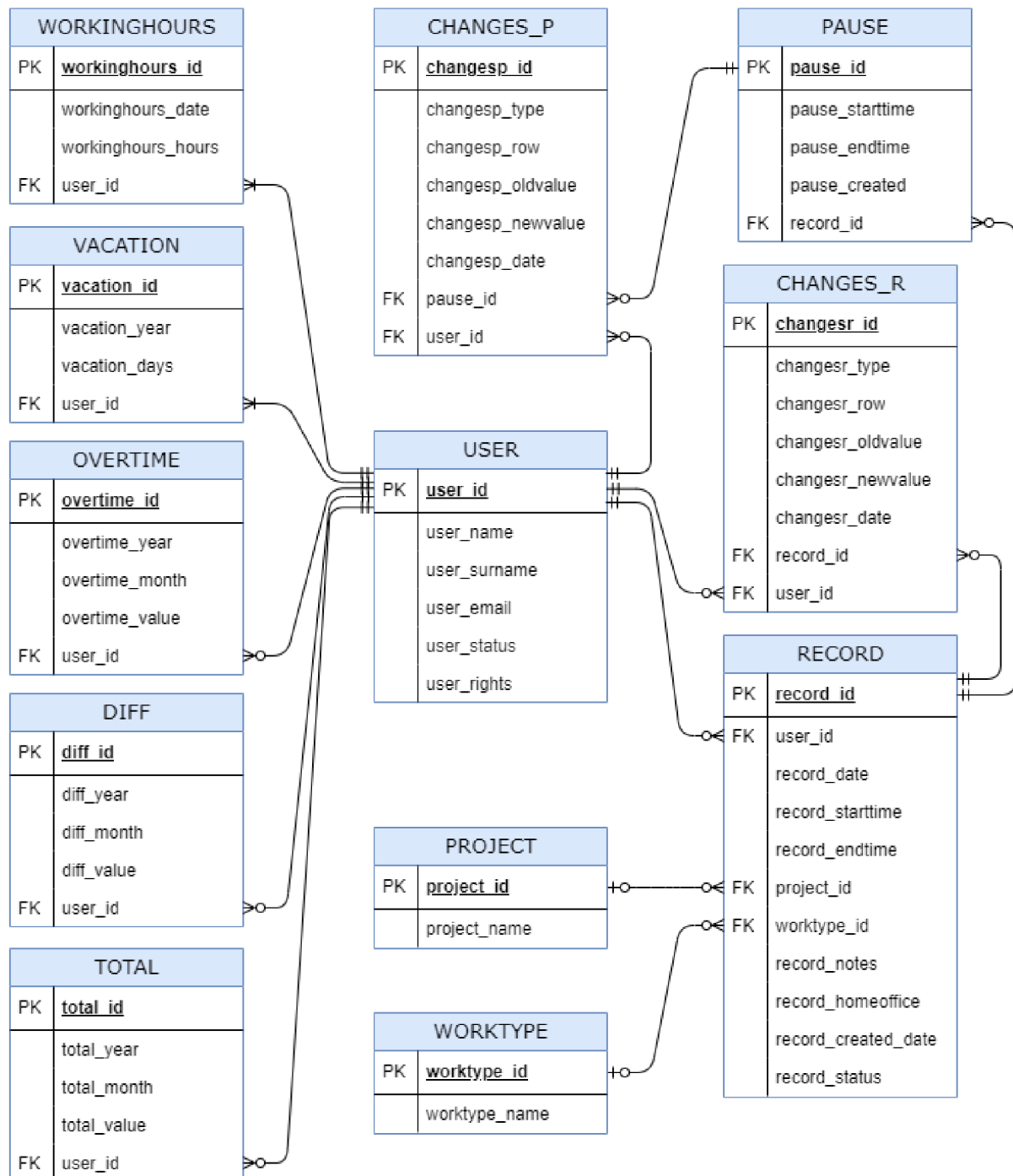
5.1 Datová vrstva

Hlavními entitami v systému pro záznam docházky uživatelů jsou RECORD a USER. Vztahy mezi těmito entitami zahrnují kardinalitu 1:N, kde každý uživatel může mít více záznamů docházky a každý záznam docházky je přiřazen právě jednomu uživateli.

Tabulka RECORD, jejíž funkcí je zaznamenávat jednotlivé řádky docházky všech uživatelů, obsahuje atributy jako ID záznamu, které je primárním klíčem, dále datum, čas zahájení docházky a její konec, poznámky k záznamu v docházce, informaci, zda se jedná o práci z domova, datum vytvoření záznamu, status, zda je záznam otevřený, připravený ke kontrole, uzamčený, tudíž ho už nelze upravovat, či odstraněný. Obsahem jsou také tři cizí klíče. Jedním je ID uživatele v tabulce USER, dalším je ID projektu z tabulky PROJECT a posledním je ID typu práce z tabulky WORKTYPE. Poslední dvě tabulky mají kromě atributu ID i atribut s názvem.

Záznam v tabulce PAUSE vzniká ve chvíli, kdy uživatel zahájí přestávku. Kromě ID pauzy obsahuje atributy jako čas zahájení přestávky a čas jejího ukončení, datum vytvoření záznamu a také cizí klíč ID záznamu z tabulky RECORD.

Další tabulkou jsou dvě tabulky CHANGES pro udržení přehledu úprav v záznamu docházky a přestávek. Její atributy jsou ID změny, typ změny jako vytvoření, úprava nebo odstranění záznamu spolu s registrovanou změnou, tedy název změněného řádku a jeho stará či nová hodnota, datum změny a cizí klíče jako ID záznamu z tabulky RECORD a ID uživatele, který změnu provedl, z tabulky USER.



Obrázek 13: ER diagram docházkového systému

K předchozí tabulce se váže příloha A, která popisuje vztahy mezi entitami.

Tabulka USER obsahuje ID uživatele neboli jednoznačný identifikátor GUID, který je uživateli přiřazen ve chvíli, kdy je mu vytvořen účet v Microsoft Entra ID. Dále zobrazuje jeho jméno, příjmení a email. Atributem je také jeho status, tedy zda se jedná o aktivního uživatele nebo uživatele, který ve společnosti už nepracuje, ale přesto se jeho docházka stále dochovává. Atribut oprávnění určuje, jaké má uživatel oprávnění v rámci aplikace docházka. K uživateli se vážou další tabulky.

Tabulka WORKINGHOURS udává úvazek uživatele, tedy kolik by měl odpracovat hodin za den. Obsahuje ID, datum, od kdy nastavený úvazek platí, hodiny, tedy při klasickém úvazku bude tento atribut obsahovat číslo 8, a cizí klíč, kterým je ID uživatele z tabulky USER, pro kterého je úvazek nastavený.

Tabulka VACATION zobrazuje, na kolik dnů dovolené má uživatel během roku nárok. Obsahuje ID, rok, dny s klasickým předpokladem 20 dnů dovolené za rok a cizí klíč ID uživatele z tabulky USER.

Tabulka DIFF zaznamenává, jaký je rozdíl vznikl uživateli v počtu odpracovaných hodin za měsíc od předpokládaného počtu odpracovaných hodin za měsíc. Obsahuje ID rozdílu, rok, měsíc, hodnotu rozdílu a cizí klíč ID uživatele.

Tabulka OVERTIME doplňuje odpracovaný počet hodin o hodnotu rozdílu v počtu odpracovaných hodin za daný měsíc v případě, kdy přesčasy byly vyplaceny místo automatického převodu do dalšího měsíce a případného vybrání si přesčasů jako volna. Tedy kromě ID zahrnuje i hodnotu jako počet hodin vyplacených přesčasů. Cizím klíčem je ID uživatele.

Tabulka TOTAL zaznamenává celkový počet hodin odpracovaných za konkrétní měsíc a aktualizuje se s každou provedenou operací nad záznamem. Díky tomu je možné získat konkrétní data z tabulky pro výpočet rozdílů v počtu odpracovaných hodin, aniž by bylo potřeba provádět složitější sčítání odpracovaných hodin celkem ze všech záznamů uživatele z databáze. Obsahuje ID celkového počtu odpracovaných hodin za měsíc, rok, měsíc, hodnotu součtu a cizí klíč ID uživatele.

5.2 Uživatelské rozhraní a fungování aplikace

První verze aplikace docházkového systému posloužila jako interaktivní prototyp, který měli možnost cíloví uživatelé vyzkoušet. Na základě toho poskytli svoji zpětnou vazbu k návrhu a uzpůsobení aplikace, případně navrhli další funkce, které by v aplikaci uvítali.

5.2.1 Docházka

Hlavní stránka pro záznam docházky „Docházka“ umožňuje uživateli zobrazit si vlastní docházku ve vybraném roce a měsíci. Výchozí zobrazení je vždy aktuální měsíc. Tabulka docházky zahrnuje datum, čas práce na projektu odkdy dokdy, přestávky, zvolený projekt a typ práce, případné poznámky a zaškrtačací políčko, zda se jedná o práci z domova.

Uživatel si může záznamy z tabulky upravovat nebo je mazat a zároveň přidávat záznamy nové. Nový záznam může být přidán pouze s datem, které odpovídá roku a měsíci zvolenému pro aktuálně zobrazené záznamy docházky. V opačném případě je tlačítko pro přidání neaktivní. Toto omezení má zajistit, aby uživatel, pokud si prohlíží docházku z jiného než aktuálního měsíce a rozhodne se přidat nový záznam, byl uvědomen o vytváření záznamu v odlišném měsíci. Díky této funkci je nový záznam vždy automaticky zobrazen v aktuálně vybraném měsíci, což minimalizuje riziko vytvoření záznamu v nesprávném měsíci v případě chybného datumu. Výchozí hodnota data je vždy nastavena na aktuální datum, takže při zobrazení jiného měsíce je možnost přidání nového záznamu neaktivní, dokud uživatel nezmění datum tak, aby odpovídal zobrazenému měsíci.

Jednou z požadovaných funkcí je možnost k jednomu záznamu přidávat více přestávek. Při přidání nové přestávky je vždy potřeba tuto přestávku potvrdit, aby se přidala do seznamu, nebo vybírat již vytvořené přestávky k záznamu a upravovat je. Vždy je ale důležité provedenou úpravu potvrdit. Pokud chce uživatel přestávku odstranit, vybere ji ze seznamu, vymaže její obsah a potvrdí. K vyplnění přestávky nejsou vždy potřeba oba časy, tj. čas začátku a konce, ale tyto informace lze doplnit i později.

Tlačítka pro úpravu nebo odstranění záznamu jsou deaktivované, dokud není zvolený záznam. Pro úpravu nebo smazání záznamu uživatel klikne na vybraný záznam ze zobrazované docházky, což načte jeho obsah do formuláře. Během úpravy nelze vytvářet nové záznamy. Pokud se uživatel rozhodne, že změnu neprovede, přičemž požaduje čistý formulář pro zadání nového záznamu, může využít tlačítko „Cancel“. To slouží nejen ke zrušení právě vybraného záznamu, ale také k rychlému vynulování formuláře.

Ve chvíli, kdy má uživatel za daný měsíc docházku řádně doplněnou, může ji potvrdit tlačítkem „Completed“. Díky tomu správce ví, že je docházka ze strany uživatele dokončená, a může s ní nadále pracovat například pro výpis pro mzdy nebo výkaz práce. Toto tlačítko má pouze informativní funkci pro správce. Při opětovné změně v docházce v daném měsíci se status resetuje, proto je potřeba potvrzení provést znovu.

Pokud jsou v záznamu doplněny veškeré potřebné časy a jsou správně, je v přehledu zobrazený celkový čas strávený na projektu. Pokud tento čas uvedený není, je to informace pro uživatele, že u záznamu buď nemá uvedené veškeré časy, nebo jsou časy mezi sebou nevalidní, jako je například čas ukončení dříve než čas zahájení, vytvoření přestávek mimo čas trvání záznamu nebo překrývání se časů přestávek v případě, kdy jich záznam obsahuje více. Typ záznamu „Vacation“ a „Sickness“ nemají v záznamu uvedené časy, jelikož jsou počítány automaticky jako jeden den podle nastaveného úvazku uživatele na den.

The screenshot shows the 'AttendanceSystem' interface. At the top, there are navigation tabs: 'Attendance' (selected), 'Overview', 'Management', 'Export', 'Settings', and 'O aplikaci'. Below the navigation, there are filters for 'Year' (2024) and 'Month' (May). There are buttons for 'Start Project', 'Start Pause', 'End Pause', and 'End Project'. Two radio buttons are visible: 'Project 1 - 08:45' (unselected) and 'Project 2 - 15:21' (selected).

Date	StartTime	EndTime	PauseStart	PauseEnd	TotalBreak	TotalHours	Project	WorkType	Note	HomeOffice
02. 05. 2024	06:15	07:00			0:00	0:45	2301 - Dr. Oetker	Time on way	Služební auto	
02. 05. 2024	07:00	14:00	09:40	11:00	1:20	5:40	2301 - Dr. Oetker	Programming onsite		
02. 05. 2024	14:00	14:45			0:00	0:45	2301 - Dr. Oetker	Time on way	Služební auto	
06. 05. 2024					0:00	0:00		Sickness		
10. 05. 2024					0:00	0:00		Vacation		
13. 05. 2024	08:45				0:00	###				
13. 05. 2024	15:21				0:00	###				

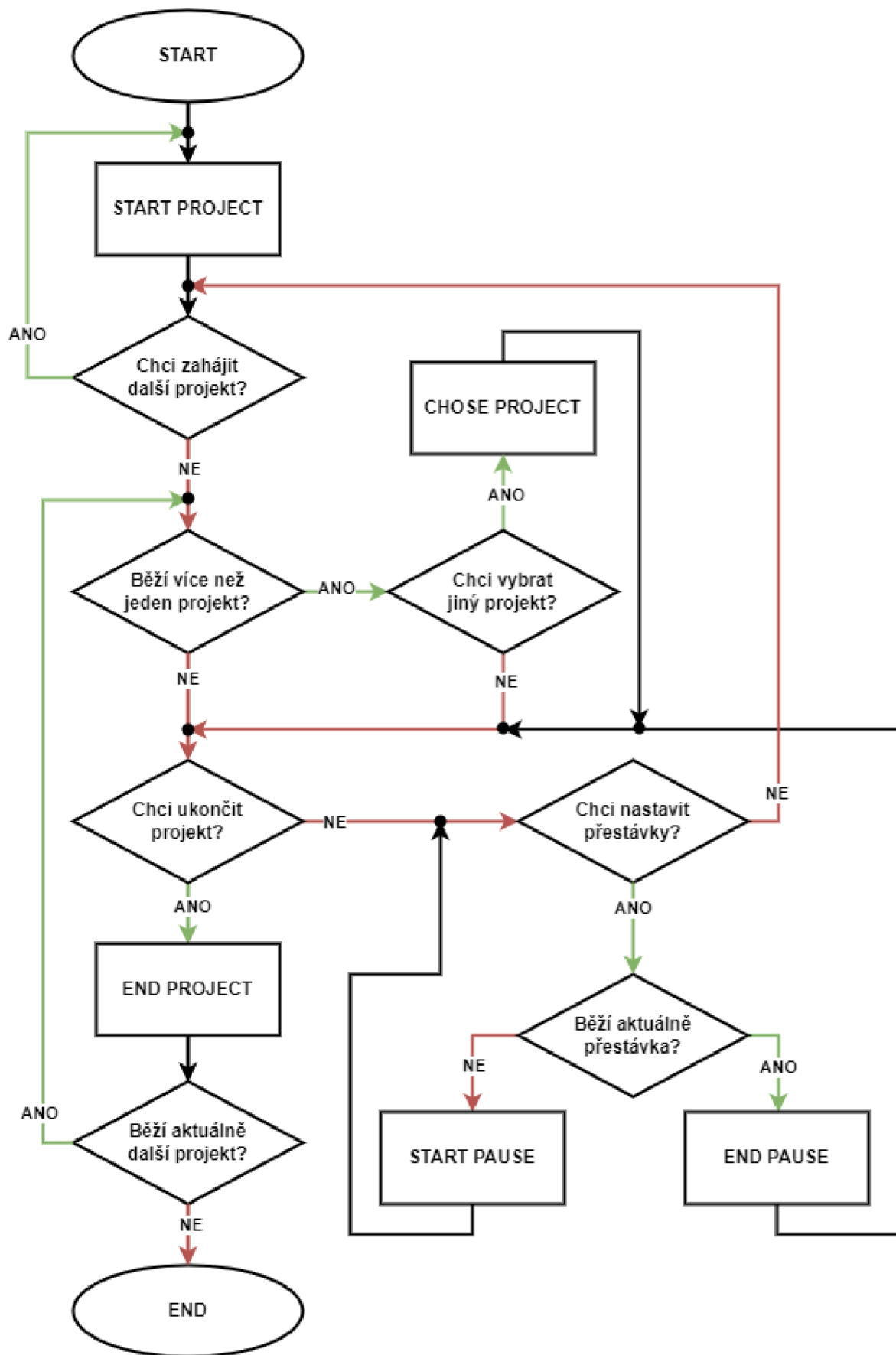
Below the table is a form for creating a new attendance record. It includes fields for 'Date' (13. 05. 2024), 'StartTime', 'EndTime', 'Pause' (with a '+' button), 'StartPause', 'EndPause', 'Project' (Start typing project), 'WorkType' (Start typing worktype), 'Note', and 'HomeOffice' (checkbox). At the bottom of the form are buttons: 'Create', 'Update', 'Delete', 'Cancel', and 'Complete'.

Obrázek 14: Záložka "Docházka" pro záznam docházky přihlášeného uživatele

Kromě klasického záznamu docházky vyplněním formuláře a přidáním dat do zobrazení může uživatel pro rychlejší zaznamenávání příchodu do práce, přestávek a odchodu z práce použít také tlačítka pro řízení aktuální práce. Jedná se o tlačítka „Start Project“, „End Project“, „Start Pause“ a „End Pause“. Čas docházky se přidá do seznamu, ale je nutné tento záznam později ještě upravit přiřazením projektu a typu práce z načteného seznamu projektů a typu práce. V seznamu právě otevřených záznamů se zobrazují pouze ty, které jsou zaznamenávány v aktuální den. Uživatel má možnost si zvolit projekt, pro který chce volbu provést, v případě, že aktuálně v jeden čas běží více projektů.

- Jedná se například o variantu, kdy zaměstnanec při příchodu do práce zahájí práci na jednom projektu, ale během toho bude muset řešit nečekaný servis pro zákazníka u jiného projektu. V takovém případě zahájí pauzu u prvního projektu a následně zahájí nový projekt, kterým je servis druhého projektu. Po ukončení tohoto druhého projektu může uživatel ukončit pauzu u prvního projektu a pokračovat v práci na něm.
- Může nastat i situace, kdy vykonávaná práce probíhá na dvou projektech paralelně, a to v případě, kdy je například zaměstnanec fyzicky v jiné firmě zákazníka pro případ, že by byla potřeba rychlý servis v případě nečekané poruchy (například při rozjezdu nově naprogramované linky). Tím má už jeden projekt v docházce zahájený a aktivní, ale během chvíle, kdy pouze čeká, zda nastane nějaká porucha, může mezitím pracovat na dalším projektu pro jiného zákazníka. Tím má aktivní dva projekty v jednu chvíli. V docházce musí být zapsané oba kvůli následnému výkazu práce zákazníkovi, ale pro výpis pro mzdy se počítá pouze jeden čas.

Díky tomuto způsobu zaznamenávání práce na projektu se docházka mírně odlišuje od klasických docházkových systémů, u kterých není řešeno, na kterém projektu pro zákazníka zaměstnanec aktuálně pracuje, ale u kterých pouze zaznamenává příchod a odchod do práce a pravidelnou 30minutovou přestávku, která musí existovat alespoň po 4 hodinách, nejpozději však po 6 hodinách. Postup záznamu práce na jednom nebo více projektech v jeden čas pomocí tlačítek je uveden v následujícím vývojovém diagramu.



Obrázek 15: Vývojový diagram pro záznam práce na projektech pomocí tlačítek v docházce

5.2.2 Statistiky

V části „Statistiky“ si může každý uživatel zobrazit přehled k jeho docházce. Statistiky jsou získány z databáze na základě ID právě přihlášeného uživatele.

Tabulka 9: Přehled zobrazovaných statistik pro zaměstnance

Statistika	Popis	Priorita výpočtu
Celkem	Celkový počet odpracovaných hodin za zvolený měsíc	1
Pracovních dnů	Počet pracovních dnů za daný měsíc	5
Rozdíl	Počet hodin, který je potřeba během měsíce odpracovat	10
Rozdíl z minulého měsíce	Rozdíl v počtu hodin z minulého měsíce	9
Dovolená	Počet dnů čerpané dovolené ve zvoleném měsíci	3
Nemocenská	Počet dnů čerpané nemocenské ve zvoleném měsíci	4
Svátky + víkendy	Počet odpracovaných hodin během svátků, soboty a neděle ve zvoleném měsíci	8
Čas na cestě	Počet hodin strávených na cestě v daném měsíci	7
Úvazek	Úvazek uživatele ve zvoleném měsíci v hodinách za den	2
Zbývající dovolená	Zbývající počet dnů dovolené v daném roce	6

1. **Celkem:** z tabulky TOTAL se získají všechny záznamy uživatele za zvolený měsíc a sečte se celkový počet odpracovaných hodin. Celkový počet k danému měsíci je v tabulce vytvořen nebo aktualizován po každém přidání nebo úpravě záznamu. Pokud záznam k danému měsíci v databázi ještě neexistuje, je hodnota nula.
2. **Úvazek:** z tabulky WORKINGHOURS se získají všechny záznamy shodné s ID uživatele a podle datumu v záznamu se získá správný záznam pro zobrazený měsíc, který obsahuje počet hodin, které se mají za den odpracovat. Pokud uživatel nemá nastavený zkrácený úvazek, je to obvykle 8 hodin za den.
3. **Dovolená:** z tabulky RECORD se získají všechny záznamy uživatele za zvolený měsíc, jejichž typ práce obsahuje hodnotu „Vacation“, a sečte se počet dnů, kdy byla čerpána dovolená.

4. **Nemocenská:** z tabulky RECORD se získají všechny záznamy uživatele za zvolený měsíc, jejichž typ práce obsahuje hodnotu „Sickness“, a sečte se počet dnů, kdy uživatel čerpal nemocenskou.
5. **Pracovních dnů:** vypočítá se počet pracovních dnů v měsíci, tedy bez sobot a nedělí, od hodnoty se odečtou dny, kdy jsou svátky a celkový počet dnů, kdy byl uživatel na dovolené nebo nemocenské (hodnoty získané z bodu 3 a 4).
6. **Zbývající dovolená:** z tabulky RECORD se získá počet záznamů uživatele za zvolený rok, jejichž typ práce obsahuje hodnotu „Vacation“, tedy počet dnů čerpané dovolené. Následně se z tabulky VACATION získá pro zvolený rok počet dnů, kdy má uživatel nárok na dovolenou. Od této hodnoty se odečte počet dnů již vyčerpané dovolené.
7. **Čas na cestě:** z tabulky RECORD se získají všechny záznamy uživatele za zvolený měsíc, jejichž typ práce obsahuje hodnotu „Time on way“, a sečte se počet hodin, které uživatel strávil na cestě.
8. **Svátky + víkendy:** z tabulky RECORD se získají všechny záznamy uživatele za zvolený měsíc, jejichž datum spadá pod sobotu, neděli nebo den, kdy byl svátek, a tento počet hodin se sečte.
9. **Rozdíl z minulého měsíce:** po uplynutí a uzamknutí měsíce se do tabulky DIFF ukládá záznam s počtem hodin, které vznikly jako rozdíl od plánovaného počtu hodin k odpracování v daném měsíci. Uživatel plánovaný počet hodin mohl přesáhnout nebo ho nemusel splnit, tudíž hodnota může být kladná i záporná. Hodnota se automaticky přenáší do dalšího měsíce. Pokud je ale kladná a byla proplacena, počet proplacených hodin se uvede do tabulky OVERTIME. Podle aktuálně zvoleného měsíce se z databáze z tabulky DIFF získá rozdíl za předchozí měsíc, a pokud k získanému záznamu existuje i záznam v tabulce OVERTIME, je získaný rozdíl snížen o hodnotu získanou z této tabulky. Pokud v tabulce DIFF za předchozí měsíc ještě neexistuje záznam, jsou veškeré výpočty pro získání rozdílu prováděny od hodnoty posledního uloženého měsíce v tabulce DIFF. Pokud pro daného uživatele v tabulce DIFF ještě neexistuje žádný záznam, je rozdíl nula v měsíci předcházejícím datumu, kdy byl uživateli zahájen úvazek, získaný z tabulky WORKINGHOURS.
10. **Rozdíl:** získaný počet pracovních dnů za měsíc (bod 5) se vynásobí hodnotou úvazku (bod 2) a od výsledku se odečte rozdíl z minulého měsíce (bod 9) a celkový počet hodin odpracovaných za zvolený měsíc (bod 1). Tím se získá počet hodin, který je potřeba v daném měsíci ještě odpracovat.

The screenshot shows the 'AttendanceSystem' interface with the 'Overview' tab selected. The 'Statistics' section displays a table of metrics for the year 2024 and month of April. The table includes the following data:

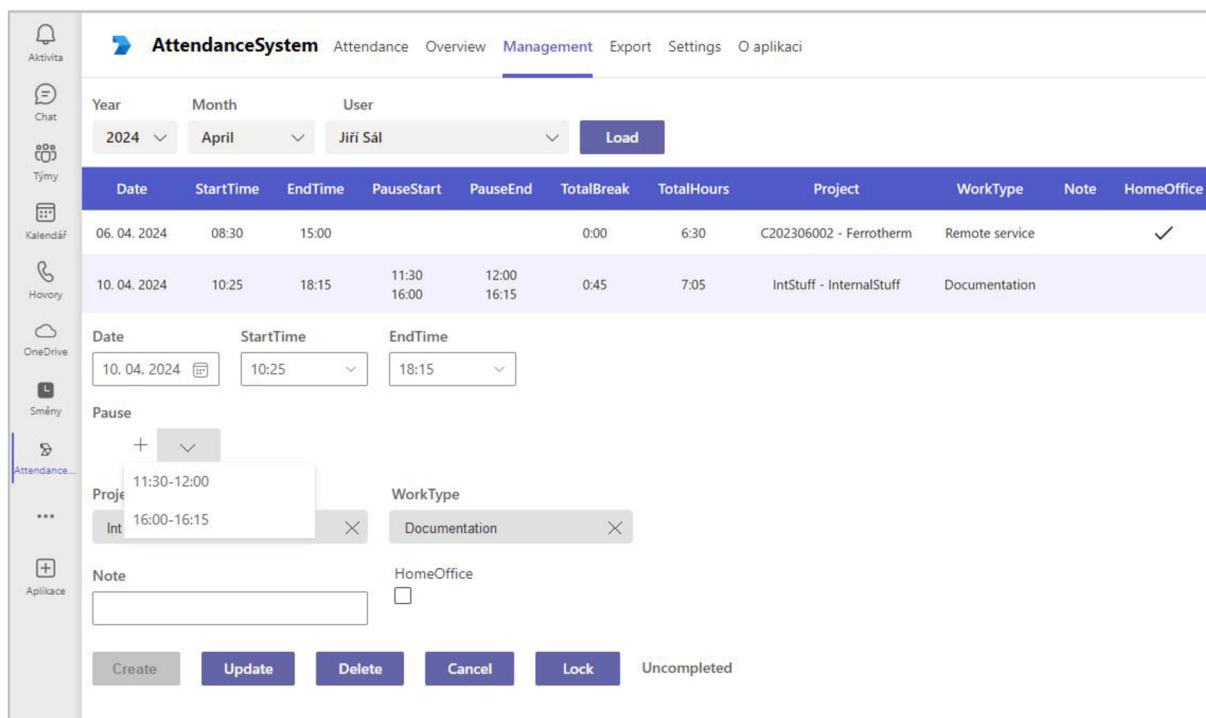
Metric	Value
Total hours	30.9
Working days in month	19
Diff (hours)	-7.1
Last month diff (hours)	0
Vacation (days)	1
Sickdays (days)	1
Holiday + weekends (hours)	0
Time on way (hours)	2
Working hours per day	2
Remaining vacation per year (days)	13

Obrázek 16: Záložka "Statistiky" pro zobrazení přehledů pro uživatele

5.2.3 Management

Záložka „Management“ představuje možnost správy docházky ostatních uživatelů pouze pověřeným správcem, který si tuto záložku může zobrazit na rozdíl od klasických uživatelů. Správce si může zvolit rok, měsíc a uživatele, jehož docházku chce spravovat. Může zkontrolovat, zda jsou všechny záznamy v pořádku a obsahují všechny nezbytné informace. V případě potřeby může záznamy přidat, upravit nebo odstranit. Veškerá změna je zaznamenávána v databázi.

Je možné zobrazit také statistiky vybraného uživatele. V případě, kdy správce provádí finální kontrolu docházky uživatele za předešlý měsíc, může následně záznamy za tento měsíc uzamknout. To se provádí z důvodu, aby zaměstnanec již zpětně nemohl provádět další úpravy v docházce, a ta tak byla připravená na generování dokumentů. V případě nutnosti je možné docházku opět odemknout, provést změny a docházku zase zamknout.



Obrázek 17: Záložka "Management" umožňuje správu docházky uživatelů správcem

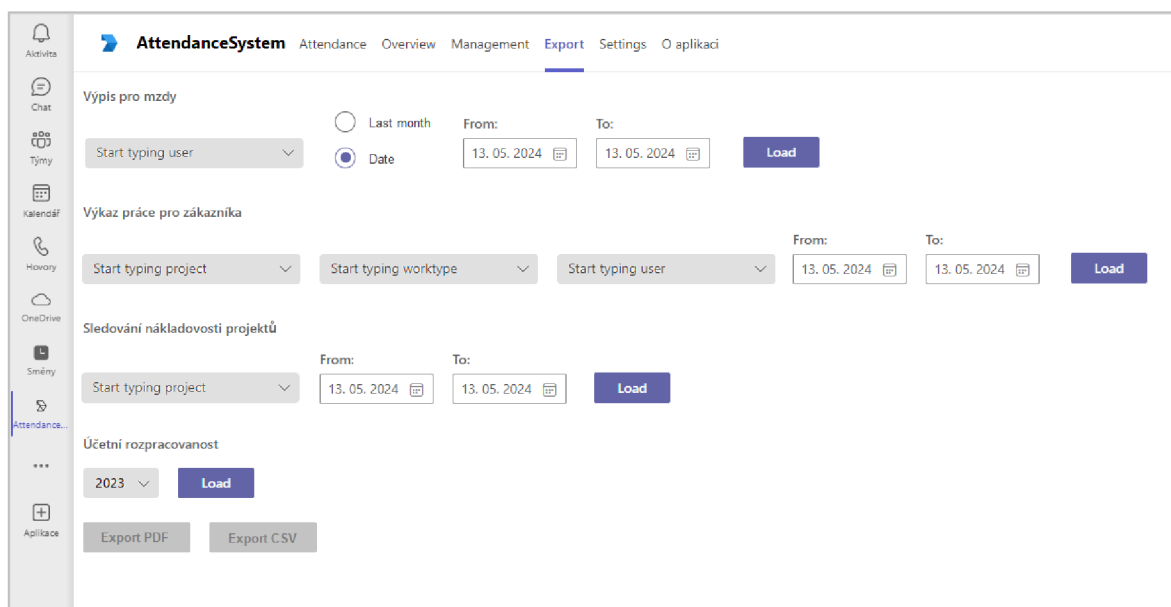
5.2.4 Export

V části „Export“ lze data z databáze správcem kdykoliv různě exportovat podle potřeby na základě vybraných kritérií.

- Výpis pro mzdy obsahuje možnost navolit uživatele, buď jednoho, více, nebo všechny. Výpis se automaticky vygeneruje za poslední měsíc či je možné zvolit konkrétní rozsah data.
- Výkaz práce pro zákazníka umožňuje navolit konkrétní projekt, případně typ práce na projektu. Také zahrnuje volbu konkrétních uživatelů, z jejichž docházky se má výkaz práce vytvořit, a výběr rozsahu datumu.
- Sledovanost nákladovosti projektu vyexportuje dokument s přehledem času stráveným na zvoleném projektu.
- Účetní rozpracovanost zahrnuje seznam výkazů všech projektů, které za předchozí rok ještě nebyly dokončeny.

Pro vytvoření dokumentu „Výpis pro mzdy“ je vytvořen dotaz na tabulku RECORD a PAUSE, který získá záznamy z tabulky se zvoleným datem, případně takové záznamy, jejichž měsíc je předchozí od aktuálního měsíce. Podmínkou je také záznam se shodným ID zvoleného uživatele. To se získá z tabulky USER podle jména zvoleného uživatele. Ke každému záznamu je také přiřazen projekt a typ práce z tabulky PROJECT a WORKTYPE. Pokud existují záznamy, které se časově překrývají, počítá se do celkového počtu odpracovaných hodin tato část pouze jednou.

Aby bylo možné vytvořit dokument „Výkaz práce pro zákazníka“, pracuje se se všemi pěti tabulkami. Nejprve se získá ID zvoleného projektu z tabulky PROJECT a ID zvolených typů práce z tabulky WORKTYPE, pokud byly vybrány. Při volbě uživatelů se také získá jejich ID z tabulky USER. Následně je vytvořen dotaz na tabulku RECORD, který obsahuje všechna získaná ID a zvolený datum. Na základě získaných záznamů se zjistí, zda existují přiřazené pauzy v tabulce PAUSE podle ID záznamu z tabulky RECORD, a tyto záznamy se také přiřadí. Díky tomu se propočítá čas strávený na projektu v jednotlivém záznamu, ale i celkem za vybrané období.



Obrázek 18: Záložka "Export" pro generování dokumentů na základě dat z databáze

„Sledování nákladovosti projektu“ zahrnuje získání ID zvoleného projektu z tabulky PROJECT a následně vytvoření dotazu na tabulku RECORD pro získání odpovídajících záznamů se shodným ID projektu, doplněných o záznamy z tabulky PAUSE. Záznamy zahrnují také jméno uživatele z tabulky USER a typ práce z tabulky WORKTYPE.

Seznam projektů na SharePointu obsahuje atribut s názvem „Dokončeno“, který slouží k potvrzení toho, že byl projekt dokončen. V takovém případě lze vygenerovat dokument „Účetní rozpracovanost“. Ze seznamu projektů na SharePointu se získá ID takových projektů, které mají hodnotu „Dokončeno“ nastaveno na „*false*“. Z databáze se následně získají záznamy z tabulky RECORD, které obsahují ID nedokončených projektů v předcházejícím roce.

Všechny vybrané záznamy se jako seznam vyexportují do souboru PDF nebo CSV.

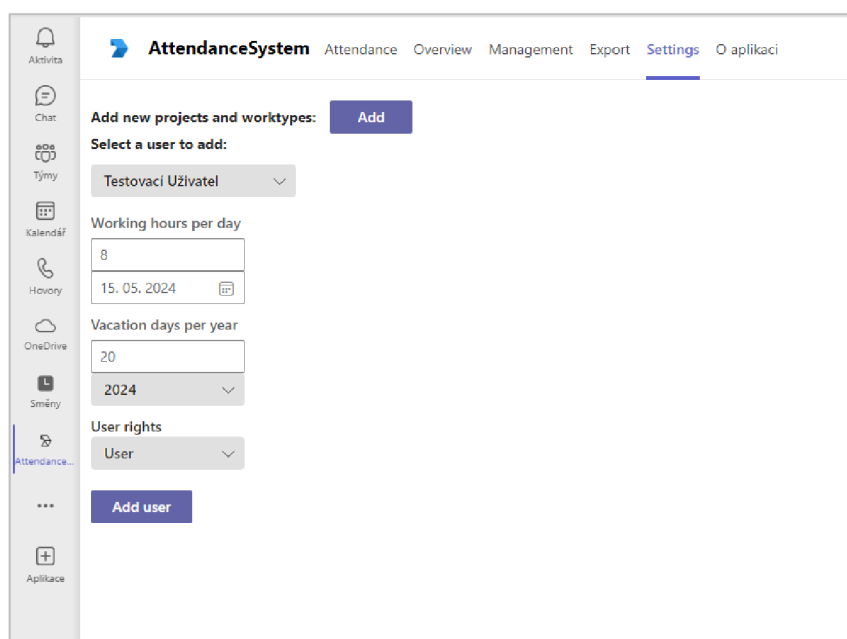
5.2.5 Nastavení

Do části „Nastavení“ v aplikaci má přístup pouze správce. V této části může provádět nastavení pro uživatele, projekty nebo typ práce. Pokud chce správce přidat do databáze nového uživatele pro záznam docházky, může nechat z Microsoft Entra ID načíst uživatele, kteří ještě nejsou uloženi v databázi. U nových uživatelů potvrdí, případně změní typ jeho úvazku a počet dnů dovolené za rok. Tato nastavení může kdykoliv v průběhu měnit. Uživatele s vazbou na existující záznamy z databáze nelze odstranit, ale lze je zneaktivnit. Pokud například byly uživateli proplaceny přesčasy, může správce v této části potvrdit, kolik hodin bylo uživateli proplaceno. Dále je zde možné z existujícího seznamu projektů načíst nové projekty a ukládat je do databáze, stejně tak jako upravovat seznam pro typ práce.

Při kliknutí na záložku se načte seznam uživatelů z Microsoft Entra ID a zkontroluje se se seznamem uživatelů uložených v databázi v tabulce USER. Výsledkem jsou uživatelé, kteří ještě nejsou uloženi v databázi. Při volbě jednoho z nich se o něm z Microsoft Entra ID získají podstatné informace a automaticky se předvyplní tabulka WORKINGHOURS s klasickým úvazkem 8 hodin denně a tabulka VACATION s plánem 20 dnů dovolené za rok. Tyto hodnoty je možné upravit.

Také se vybere typ oprávnění pro uživatele, který je svázán s tabulkou RIGHTS. Po kliknutí na tlačítko „Add user“ je vytvořen nový uživatel v tabulce USER. Tabulka pro záznam dovolené uživatele je pro každý další rok automaticky vytvořena se stejnými hodnotami. Správce je opět může později měnit stejně tak jako nastavovat, zda je účet uživatele stále aktivní.

Na SharePointu existuje seznam projektů a typu práce, který se neustále rozšiřuje. Při kliknutí na tlačítko „Update projects and worktype“ dojde k přidání nových projektů do tabulky PROJECT a nových typů práce do tabulky WORKTYPE, pokud ještě neexistují. Aktualizaci těchto seznamů je možné provádět automaticky jedním tlačítkem.



Obrázek 19: Záložka "Nastavení" pro základní nastavení a práci s databází

Při použití aplikace se nejprve získá ID přihlášeného uživatele a podle nastaveného oprávnění v tabulce USER v databázi získá přístup pouze k určitým záložkám aplikace. Během používání aplikace si systém pamatuje jeho ID, načítá pouze záznamy, které obsahují jeho ID, a při každé změně provedené v aplikaci se jeho ID k záznamu automaticky ukládá. Uživatel, který není zaveden v databázi, nemá k docházce přístup.

6 Tvorba a integrace SPFx aplikace

Vytvoření samostatné webové aplikace a následně její implementace do Microsoft Teams je vhodné spíše v případě, pokud se jedná o aplikace, která nijak nepracuje a nekomunikuje s prvky uvnitř Teams nebo SharePointu. U takové aplikace bylo nutné vytvářet autentizaci. Vytvoření aplikace v Power Apps je omezené funkcemi a logikou a také prací s datovými zdroji. Propojení Power Apps s databází je prémiovou funkcí a propojení se seznamy na SharePointu není dynamické. Každý nový datový zdroj je potřeba ručně přidat. Tudiž pokaždé, kdy například dojde k přidání nového uživatele, je i nutné mu vytvořit nový seznam pro uchovávání dat z docházky a tento zdroj ručně propojit s Power Apps. Vhodným způsobem tvorby a integrace aplikace docházkového systému pro MS Teams je tedy způsob pomocí SPFx.

Veškeré postupy, příkazy a potřebné zdrojové kódy k této kapitole jsem sepsala v příloze B tohoto dokumentu.

6.1 SharePoint Framework Extensions

SharePoint Framework (SPFx) je evolucí vývoje SharePointu, která odpovídá potřebám moderního prostředí. Původní architektura založená na serverových úpravách pomocí C# fungovala dobře v prostředí s pouze jedním podnikem, ale nedokázala efektivně škálovat do cloudu, kde běží několik tenantů na stejných serverech. Z toho důvodu Microsoft představil alternativní modely, jako je vkládání JavaScriptu na straně klienta a doplňky SharePoint.

První varianta umožňuje vkládání JavaScriptu do webové části Script Editor a jeho spuštění při zobrazování stránek. Tento přístup nabízí interakci s ostatními prvky na stránce a je relativně snadno použitelný. Ovšem má omezení v poskytování konfiguračních možností pro uživatele a zahrnuje bezpečnostní riziko úprav skriptu, které může poškodit funkčnost webové části. Navíc v samoobslužných kolekcích webů je aktivní funkce „*NoScript*“, která blokuje spuštění skriptů webové části na stránce.

Na serveru SharePoint Server 2013 existuje možnost řešení zvané jako model doplňku/části aplikace, která běží i na webech s aktivní funkcí „*NoScript*“. Tento přístup vytváří prvek iFrame, který obsahuje a spouští skutečné prostředí mimo aktuální stránku, což zvyšuje jeho spolehlivost. Avšak iFrame se stává pomalejší než webová část Script Editor, protože vyžaduje další požadavky na jinou stránku a má omezené možnosti vytváření responzivních návrhů a úprav CSS. [17]

V reakci na předchozí kontextuální změny se jako dalším krokem vývoje SharePointu stal SharePoint Framework, který přinesl moderní přístup založený na JavaScriptu a REST API voláních.

- Běží v kontextu aktuálního uživatele a připojení v prohlížeči. Neexistují žádné iFrame prvky pro přizpůsobení (JavaScript je vložen přímo do stránky).
- Ovládací prvky jsou vykresleny v DOM části stránky a jsou snadno přístupné a responzivní.
- Umožňuje vývojářům přístup k životnímu cyklu jako renderování, načítání, serializace a deserializace, změny konfigurace a další.
- Je to nezávislé na frameworku. Lze použít jakýkoli framework JavaScriptu, jako React, Handlebars, Knockout, Angular nebo Vue.js.
- Vývojářský „*toolchain*“ je založen na populárních open-source klientských vývojových nástrojích, jako jsou NPM, TypeScript, Yeoman, Webpack a Gulp.
- Výkon je spolehlivý.
- Koncoví uživatelé mohou používat řešení SPFx na straně klienta, která jsou schválena správci tenantů (nebo jejich delegáty) na všech webech, včetně samoobslužných týmových, skupinových nebo osobních webů.
- Webové části SPFx lze přidat na klasické i moderní stránky, použít k rozšíření Microsoft Teams, Outlook a dalších aplikací Microsoft 365 a k Microsoft Viva Connections.

Běh programu je vylepšen ve webové části Script Editor. Zahrnuje robustní klientské API, objekt HttpClient, který zpracovává ověřování pro SharePoint a Microsoft 365, kontextové informace, snadnou definici, konfiguraci vlastností a další. [19]

6.2 Příprava prostředí

Pro vývoj se používá vývojové prostředí Visual Studio Code. Je potřeba mít nainstalované Node.js verze 14, v16 nebo v18, protože vyšší verze prozatím nejsou kompatibilní. Také je vhodné nainstalovat SharePoint Online Management Shell. Ten se využije jako konzole pro přihlášení se jako tenant na SharePoint a povolení CDN (Content Delivery Network) pro ukládání statických obsahových prostředků (obrázky, JS soubory, CSS atd.) do mezipaměti k prohlížečům pro zvýšení výkonu a snížení latence při načítání SharePoint stránky.

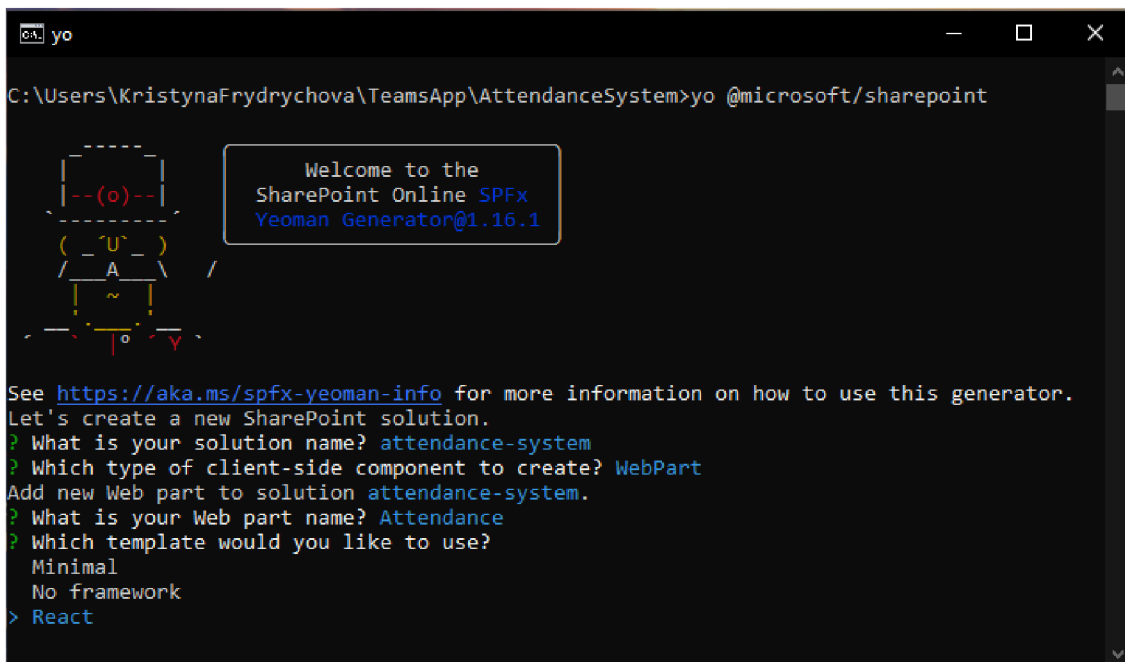
Dále je potřeba nainstalovat nástroje Yeoman a Gulp a následně i SharePoint generátor `@microsoft/sharepoint`. Yeoman vytvoří novou aplikaci. Gulp se používá k sestavení, náhledu a testování projektu. Správce javascriptových balíčků NPM se používá pro správu skriptů.

SharePoint správce musí přidat a nastavit App Catalog pro přidávání aplikací na SharePoint v balíčku „.sppkg“. Jako Teams správce je zas potřeba povolit uživatelům přidávání vlastních aplikací nebo jen konkrétních do MS Teams, aby si novou aplikaci po integraci mohli přidat a nainstalovat.

6.3 Vytvoření základu SPFx aplikace spolu s FluentUI

V příkazovém řádku je potřeba vybrat adresář pro vytvoření nové aplikace a zde nechat aplikaci vygenerovat. SharePoint Online SPFx Yeoman generátorem je vytvořeno projektové řešení se soubory, nástroji a strukturou projektu vhodné k integraci do SharePointu. Při spuštění a vytvoření nového projektu SharePoint Framework Extensions (SPFx) se volí název aplikace, název webových částí, typ komponent a framework aplikace. V tomto případě se jako framework zvolí React pro přívětivější tvorbu uživatelského rozhraní.

Pokud je potřeba vytvořit více webových částí neboli záložek, spustí se opět stejný příkaz jako pro vytvoření aplikace, a to v adresáři, ve kterém je aplikace vytvořena. Pak se jen zvolí typ vytvoření webové části „webpart“.



```
C:\Users\KristynaFrydrychova\TeamsApp\AttendanceSystem>yo @microsoft/sharepoint

Welcome to the
SharePoint Online SPFx
Yeoman Generator@1.16.1

See https://aka.ms/spfx-yeoman-info for more information on how to use this generator.
Let's create a new SharePoint solution.
? What is your solution name? attendance-system
? Which type of client-side component to create? WebPart
Add new Web part to solution attendance-system.
? What is your Web part name? Attendance
? Which template would you like to use?
  Minimal
  No framework
  > React
```

Obrázek 20: Vygenerování SPFx aplikace pomocí Yeoman generátoru

Je nutné doinstalovat framework pro tvorbu designu FluentUI. Obsahuje sadu komponent pro vytváření moderních a přizpůsobitelných uživatelských rozhraní a utilit ze sady knihoven komponent založených na Reactu `@fluentui/React` a `@fluentui/react-northstar`. Původně byl vyvinut firmou Microsoft. Nabízí sadu vizuálních prvků, komponent a stylů, které je možné použít pro aplikace v Microsoft 365. Je postaven na designovém jazyce nazvaném Fluent Design System a jeho cílem je poskytnout jednotný a moderní vzhled a chování aplikací na různých zařízeních a platformách. FluentUI je tedy implementací těchto návrhových principů v podobě programovatelných komponent. Knihovna je často používána ve spojení s technologiemi, jako jsou React nebo Angular, a je k dispozici jako open-source projekt, což umožňuje komunitě přispívat k jejímu rozvoji.

6.4 Vytvoření manifestu

Aby bylo možné aplikaci zobrazit v Microsoft Teams, musí obsahovat soubor manifestu a ikonky. Manifest soubor ve formátu JSON obsahuje metadata, jako je ID a název aplikace a její popis, jména vývojářů, verze, odkazy na ikony a aplikaci samotnou. Jeho obsahem jsou hlavně struktury webových částí aplikace „*Tabby*“, které jsou v manifestu propojeny pomocí odkazů „*contentUrl*“ obsahujících ID těchto webových částí „*SPFx component ID*“ převzatých z manifest souborů jednotlivých webových částí. Do tohoto souboru se také přidávají oprávnění, povolené domény „*validDomains*“.

Do části „*webApplicationInfo*“ se vkládá ID zaregistrované aplikace v Azure Portal v Microsoft Entra ID, která je propojena s SPFx řešením. Výchozí registrovanou aplikací v Azure je „*SharePoint Online Client Extensibility Web Application Principal*“, proto je nutné vložit ID této aplikace. Tento krok je potřebný k tomu, aby bylo zajištěno, že jakékoli volání rozhraní API v SPFx komponentě bude úspěšně provedeno a komunikace povolena. Adresa domény „*teamSiteDomain*“ má stejného tenanta, kterého používá klient Teams, takže volání bude probíhat ve stejné doméně.

Manifest soubor je možné vytvořit ručně podle předlohy nebo si ho nechat vygenerovat v nástroji Developer Portal v MS Teams podle zadaných informací a spolu se dvěma ikonkami stáhnout, případně dopravit a vložit do SPFx řešení.

6.5 Nahrání na SharePoint

Pro nahrání aplikace na SharePoint je nejprve potřeba komprimovat obsah složky „./teams“ projektu, která obsahuje soubor „manifest.json“ a dvě ikonky „outline.png“ a „color.png“. Platným balíčkem aplikace Teams je pouze obsah této složky, ne složka celá. Komprimovaný balíček je nutné pojmenovat „TeamsSPFxApp.zip“ a nechat ho uvnitř složky „./teams“. Špatný název, umístění nebo obsah balíčku, špatné ikonky nebo špatně nastavený manifest může způsobit, že aplikace v Teams nebude vytvořena.

Následně je potřeba z celé aplikace vytvořit balíček „.sppkg“, který bude možné nahrát na SharePoint. Balíček je vytvořen běžným příkazem pro „build“ a „package“ řešení, tedy „gulp bundle –ship“ a následně „gulp package-solution –ship“. Výsledný balíček se vytvoří ve složce „./sharepoint/solution“, takže je možné ho nahrát na SharePoint do App Catalog, odkud se integruje do MS Teams.

Pro vytvoření katalogu aplikací v SharePointu je nutné mít nainstalovaný a nakonfigurovaný „App Catalog Site Collection“, což je kolekce webů, která je určena jako centrální místo pro správu a distribuci aplikací pro SharePoint. Poté je potřeba jako správce vytvořit katalog aplikací, do kterého lze přidávat různé typy aplikací, které jsou v tomto katalogu uloženy a jsou k dispozici pro uživatele v celé organizaci. Pro používání SharePoint katalogu aplikací, je nutné mít potřebné oprávnění a znalosti správy SharePointu.

Po nahrání balíčku do katalogu aplikací se analyzuje obsah balíčku řešení a v případě potřeby se zobrazí možnost publikovat řešení automaticky také v MS Teams. To zjednodušuje proces nasazení, aby bylo řešení snadno dostupné v rámci celé organizace Microsoft 365. Pokud vše proběhlo správně, aplikace by se po chvíli měla objevit v samotném Teams, odkud si už uživatelé mohou aplikaci ze seznamů aplikací pro organizaci přiřadit a nainstalovat. Ta se objeví v levém postranním panelu jako samostatná osobní aplikace. Pokud není povolené automatické přidávání vlastních aplikací, je potřeba aplikaci po nahrání balíčku do katalogu aplikací na SharePoint nejprve schválit Teams správcem na kartě oprávnění aplikací Teams.

6.6 Testování

Testování aplikace lze provádět lokálně nebo online pomocí pracovní plochy „Workbench“ nebo přímo na platformě MS Teams. To pomáhá odladit kód, otestovat fungování komponent nebo vidět rozvržení UI před tím, než se aplikace nasadí.

6.6.1 Lokálně přes Workbench

Pro testování lokálně je potřeba nejdříve přidat certifikát localhost „*gulp trust-dev-cert*“. Pro práci přes Workbench je nutné doinstalovat moduly pro zobrazení webových částí „*@microsoft/sp-webpart-workbench*“. V souboru pro konfiguraci „*config/serve.json*“ je při lokálním testování nutné změnit URL adresu na lokální „*https://localhost:5432/workbench*“, přidat port a cestu k Workbench modulům. Následně spustit server „*gulp serve*“. V prohlížeči se otevře prázdná lokální stránka s pracovní plochou, kde si vývojář může přidat svoje řešení. Ovšem je možné zde pracovat pouze s fiktivními daty a zobrazit uživatelské rozhraní webové části, ale není zde přístup ke skutečnému kontextu SharePointu.

6.6.2 Online přes Workbench

Pro přístup ke všem datům prostředí SharePoint, jako jsou seznamy, knihovny, kontext stránky atd., je pouze potřeba upravit URL adresu v konfiguračním souboru za reálnou adresu SharePoint webové stránky, ke které je uživatel přihlášený, doplněnou o adresu pracovní plochy „*/_layouts/workbench.aspx*“ a vybrat svoje řešení z místní nabídky. Nemusí se jednat o stejnou SharePoint webovou stránku, se kterou SPFx řešení pracuje, ovšem název tenantu musí být totožný. Příkladem je: „*https://{tenant_name}.sharepoint.com/sites/{sharepoint_page_name}/_layouts/15/workbench.aspx*“. Po spuštění serveru je možné upravovat kód ve vývojovém prostředí a při každém uložení se změny okamžitě projeví na webové stránce, takže ladění probíhá rychleji.

6.6.3 Lokálně přes Teams

Testovat aplikaci je možné také přímo v MS Teams, například pro kontrolu rozvržení uživatelského rozhraní, barev a dalších grafických prvků nebo pro ověření fungování složitější komunikace mezi dalšími nástroji Microsoft. Adresu v konfiguračním souboru je potřeba opět nastavit na lokální a následně spustit „*build & package*“ řešení, ovšem bez příkazu „*--ship*“, tedy „*gulp bundle*“ a „*gulp package-solution*“. Balíček se následně nahraje na SharePoint do katalogu aplikací, povolí se pro organizaci a nechá se synchronizovat s Teams. Přidaná a nainstalovaná aplikace v prostředí MS Teams se načte jen v případě, kdy je spuštěn server SPFx řešení. Po každé změně, která je provedena ve VS Code během toho, co je server spuštěn, se po uložení změny nahrají a zobrazí přímo v aplikaci na platformě Teams po znovunačtení stránky. Aplikace v MS Teams je tak určena pouze pro testování a bez spuštěného serveru samostatně nefunguje.

6.6.4 Online přes Teams

Klasická integrace aplikace do MS Teams je obvykle už určena pro ostrý provoz a způsob této integrace je popsán v podkapitole „6.4 Nahrání na SharePoint“. Aplikace je tak zpřístupněna celé organizaci a používat ji může kdokoli. Případně může Teams správce povolit aplikaci pouze pro konkrétní uživatele. Tím lze testovat aplikaci na reálných datech a v provozu, aniž by aplikaci mohl používat jiný uživatel. Druhou variantou je Microsoft 365 Developer Program, který je určen vývojářům jako testovací prostředí Microsoftu a nenarušuje tak produkční prostředí.

Pro nasazení aplikace:

```
"initialPage": "https://enter-your-SharePoint-site/_layouts/workbench.aspx"
```

Pro testování aplikace:

```
"initialPage": "https://localhost:5432/workbench",  
  "api": {  
    "port": 5432,  
    "entryPath": "node_modules/@microsoft/sp-webpart-workbench/lib/api/"  
  }  
}
```

Zdrojový kód 3: Úprava konfiguračního souboru `config/serve.json`

6.7 Konfigurace a další úpravy v aplikaci

Manifest každé webové části „`{webpart_name}WebPart.manifest.json`“ by měl obsahovat informaci o podporovaných službách pro hostování aplikace „`supportedHosts`“, jejímž obsahem jsou „`SharePointWebPart`, „`TeamsPersonalApp`, „`TeamsTab`, „`SharePointFullPage`“. To umožňuje automatické nasazení řešení do MS Teams, když je nainstalováno v katalogu aplikací SharePoint. V opačném případě by se soubor „`TeamsSPFxApp.zip`“ musel ještě nahrát na platformu Teams, aby se spároval s řešením vloženým do SP katalogu aplikací.

V souboru „`./config/package-solution.json`“ je potřeba přidat „`skipFeatureDeployment`“: `true`“, aby bylo řešení dostupné v rámci celé organizace na platformě MS Teams.

6.8 Práce s daty

Knihovna `@pnp/sp` je zaměřená pro práci s TypeScriptem, ale funguje i pro projekty vytvořené v JavaScriptu. Vhodné je nainstalovat verzi 2, jelikož verze 3 vykazuje chyby při práci se SharePoint seznamem. Pro přístup k SharePoint REST API je určen balíček `@pnp/sp` a pro přístup k Microsoft Graph API se využívá `@pnp/graph`. Knihovny jsou navrženy tak, aby bezproblémově fungovaly v rámci projektů SharePoint Framework s nastavením pouze drobných konfigurací.

6.8.1 Propojení se SharePoint pomocí SPHttpClient

Aby bylo možné přistupovat ke kontextu SharePoint webu, jako jsou seznamy, soubory, knihovny a stránky, je potřeba SPFx řešení propojit se SharePoint REST API. K tomu je od Microsoftu připravená instance `SPHttpClient`, která je už součástí SharePoint Frameworku, tudíž nejsou potřeba žádné další závislosti pro její použití. Ovšem při složitějších REST požadavcích je náchylnější k chybám.

```
this.context.spHttpClient
  .get(`${this.context.pageContext.web.absoluteUrl}/_api/web?$select=Title`
  ,
    SPHttpClient.configurations.v1,
    {
      headers: [
        ['accept', 'application/json;odata=verbose']
      ]
    })
  .then((res: SPHttpClientResponse): Promise<{ Title: string; }> => {
    return res.json();
  })
  .then((web: { Title: string }): void => {
    console.log(web.Title);
  });
```

Zdrojový kód 4: Získání názvu SP webu za použití instance `SPHttpClient`

Druhou variantou je open-source JavaScriptová knihovna `PnPjs` – `@pnp/sp`. Také poskytuje zabezpečené rozhraní API pro vytváření REST dotazů v SPFx řešení. Má mnohem jednodušší kód, jelikož jsou všechny prvky požadavků při vykonání REST požadavku již pevně zapsány, proto má i nižší riziko chyb za běhu. Má aktivně vyvíjející se komunitu, ale funguje jako další závislost, kterou je potřeba do projektu přidat a při nových aktualizacích spravovat.


```

import { Web } from "@pnp/sp/presets/all";

public async getData() {
    const web = Web(this.props.webURL);
    let items: any[] = [];
    items = await
web.lists.getByTitle("Attendance").items.select('*').get();
    console.log(items);
}

```

Zdrojový kód 5: Získání položek seznamu z SP webu pomocí knihovny PnPjs

Celý tento proces je spojený s automaticky registrovanou aplikací „*SharePoint Online Client Extensibility Web Application Principal*“ v Azure Portal, přes kterou SPFx získává oprávnění pro svá volání.

6.8.2 Propojení s Microsoft 365 pomocí MSGraphClient

Microsoft Graph funguje jako brána ke všem službám v Microsoft 365. Jedná se o přístup k velkému množství dat z různých služeb v cloudu. Jelikož má každá služba své vlastní API, existuje tak spousta frameworků a vestavných API pro přístup k různým datům. Aby byl tento přístup k datům co nejsnazší, vzniklo Graph API jako jedno API pro téměř všechny služby Microsoft 365.

Aby mohlo SPFx pracovat s daty uvnitř Microsoft 365, je nejprve potřeba povolit oprávnění aplikaci, aby k těmto konkrétním datům mohla přistupovat. V SharePoint centru pro správce v sekci „Správce rozhraní API“ (Upřesnit -> Přístup k API) je možné ze skriptů a komponent SharePoint Frameworku spravovat přístup k rozhraním API, která jsou zabezpečená službou Microsoft Entra ID. Správce může z případného seznamu žádostí čekající na vyřízení schvalovat potřebná oprávnění. Tato oprávnění je potřeba buď nejprve přidat ve správě Microsoft Entra ID na portálu Azure, nebo přímo v SPFx aplikaci.

6.8.2.1 Vytvoření oprávnění SPFx ke službám Microsoft 365

V Azure Portal ve službě Microsoft Entra ID v sekci „Registrace aplikací“ v seznamu „Všechny aplikace“ je registrovaná aplikace „SharePoint Online Client Extensibility Web Application Principal“. Po jejím otevření v sekci „Oprávnění rozhraní API“ je potřeba přidat nové oprávnění aplikace. Po rozkliknutí položky „Přidat oprávnění“ se v horní části při výběru rozhraní API volí „Rozhraní API Microsoft“.

Ze seznamu je potřeba vybrat položku „Microsoft Graph“ a následně „Delegovaná oprávnění“, jelikož pro používání aplikace bude vždy uživatel přihlášen pod svým účtem Microsoft 365 na rozdíl od druhého typu oprávnění, kdy aplikace běží jako služba na pozadí nebo samostatně bez přihlášeného uživatele.

V této části se již volí potřebné Graph API oprávnění. Například pro přístup SPFx aplikace ke čtení základních profilů všech uživatelů je potřeba v části „User“ vybrat oprávnění „User.ReadBasicAll“ nebo jakékoliv další oprávnění, se kterým je potřeba pracovat. Po kliknutí na „Přidat oprávnění“ je možné rovnou přidaná oprávnění odsouhlasit správcem „Udělit souhlas správce pro ...“, čímž je API připraveno k použití.

Nyní opět v sekci „Správce rozhraní API“ v SharePoint centru pro správce je možné toto udělené oprávnění zobrazit již ve schválených oprávněních. Pokud by v předchozím bodě nedošlo ke schválení oprávnění přímo na portále Azure, bylo by přidané oprávnění v seznamu žádostí čekající na vyřízení a udělení přístupu by bylo možné schválit i z této stránky.

Oprávnění je možné přidat i přímo v aplikaci SPFx v souboru „*package-solution.json*“, ale po nahrání aplikace do SharePoint katalogu aplikací bude potřeba požadovaná oprávnění aplikace nejprve správcem schválit opět v sekci „Správce rozhraní API“.

```
"webApiPermissionRequests": [  
  {  
    "resource": "Microsoft Graph",  
    "scope": "User.ReadBasicAll"  
  }  
],
```

Zdrojový kód 6: Oprávnění SPFx aplikace pro čtení základních profilů všech uživatelů v souboru *package-solution.json*

6.8.2.2 Práce s MS Graph API v aplikaci SPFx

Z SPFx aplikace se lze připojit k Microsoft Graph pomocí služby MSGraphClientV3, která zároveň řeší i ověřování. Do TypeScriptového interface souboru ve složce „*components*“ je potřeba importovat Graph API klienta. Následně je na něj v hlavním TypeScript souboru v metodě „*render*“ vytvořena instance. Při získání odkazu na instanci je už možné i z React souboru komunikovat s MS Graph pomocí syntaxe klientské knihovny Microsoft Graph JavaScript. [21]

Import MSGraphClientV3 v interface souboru IAttendance.ts:

```
import { MSGraphClientV3 } from '@microsoft/sp-http';

export interface IAttendanceProps {
  client: MSGraphClientV3;
}
```

Vytvoření instance MSGraphClientV3 v souboru Attendance.ts:

```
public async render(): Promise<void> {
  const element: React.ReactElement<IAttendanceProps> =
  React.createElement(
    Attendance,
    {
      client: await this.context.msGraphClientFactory.getClient('3')
    }
  );
  ReactDOM.render(element, this.domElement);
}
```

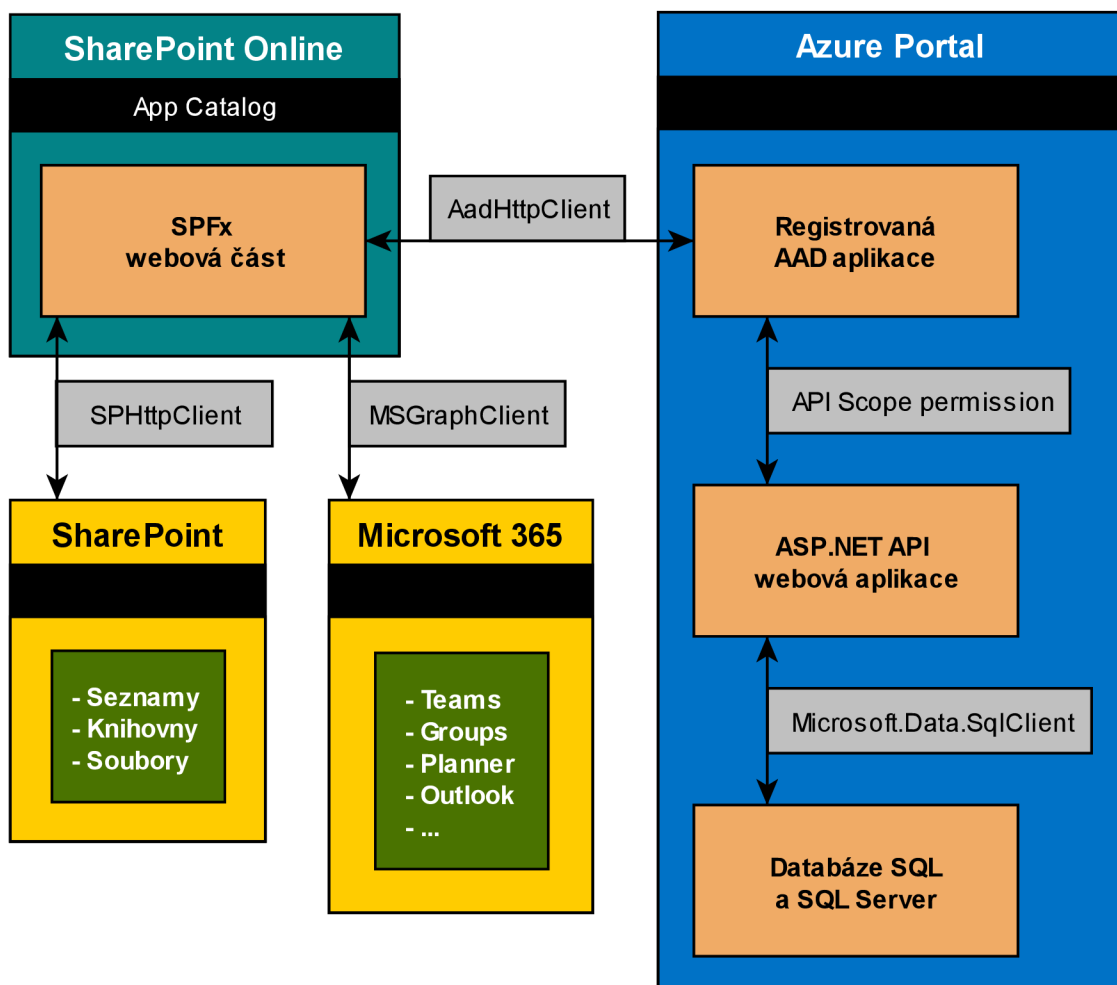
Komunikace s MS Graph API ze souboru Attendance.tsx:

```
private async loadUserDetails(): Promise<void>{
  const userInfo: any = await
  this.props.client.api('/users').version("v1.0").get();
  console.log(userInfo);
}
```

Zdrojový kód 7: Příprava prostředí pro komunikaci s Microsoft Graph API z aplikace SPFx

6.8.3 Propojení s databází Azure pomocí AadHttpClient

Aby mohla SPFx aplikace pracovat s daty z Azure SQL Database, je potřeba přidat API, které funguje jako prostředník mezi klientskou částí aplikace a databází. To zahrnuje vytvoření a nastavení databáze v Azure SQL Database, registraci aplikací pro SPFx a API v Microsoft Entra ID a nastavení oprávnění pro komunikaci s databází. V Azure App Service je potřeba vytvořit API, které obsahuje metody pro manipulaci s daty v databázi a je zabezpečeno pomocí autentizace Microsoft Entra ID. SPFx aplikace musí mít nakonfigurovaný AadHttpClient, který může používat k volání metod API v App Service při vytvoření požadavku spolu s přístupovým tokenem získaným z Microsoft Entra ID k autentizaci při komunikaci s API. To následně umožňuje zpracovat přijaté požadavky, provést odpovídající operace s databází díky knihovně Microsoft.Data.SqlClient a vrátit odpověď obsahující požadovaná data zpět do SPFx aplikace, která po obdržení odpovědi od API data zpracuje a zobrazí je uživateli ve webovém rozhraní.



Obrázek 21: Architektura aplikace

6.8.3.1 Vytvoření SQL databáze a SQL serveru v Azure Portal

Azure SQL Database je plně spravovaná relační databáze nabízená v rámci cloudové platformy Microsoft Azure. Microsoft se stará o údržbu a aktualizaci serveru, poskytuje zabezpečení dat šifrováním a možnostmi zálohování a umožňuje vytvářet zálohy a automatickou replikaci dat mezi datovými centry. Uživatelé si mohou upravit výkon a kapacitu databází podle svých aktuálních potřeb, což umožňuje rychlé reagování na změněné podmínky a platbu jen za to, co opravdu využívají. Azure SQL Database je dostupná pro spoustu aplikací, včetně webových aplikací, podnikových systémů, mobilních aplikací a analytických úloh.

Při vytváření nové databáze ve službě Azure SQL Database je potřeba nejprve vytvořit server služby SQL Database, zvolit metodu ověření pro přístup k serveru (ověření přes SQL nebo Microsoft Entra nebo obojí) a nastavit správce nebo údaje pro přihlášení správce k serveru.

V případě využití jednoúčelové SQL databáze s přerušovaným a nepředvídatelným využitím na úrovni bezserverových výpočetních prostředků dochází díky automatickému škálování výpočetních prostředků k fakturaci pouze využitých výpočetních prostředků po sekundách. Tedy náklady na výpočty za jedno virtuální jádro za sekundu, na základě kombinace využití procesoru a paměti, jsou 0,000159 dolarů a v závislosti na procentu využití tohoto jednoho virtuálního jádra se náklady za sekundu snižují. Náklady za 1 GB uložení na měsíc jsou 0,18 dolarů. Navíc po dobu trvání předplatného je možné využít bezplatnou službu Azure SQL Database s 100 000 sekundami výpočetních prostředků a 32 GB uložení pro jednu databázi. Po dosažení bezplatného limitu je možné nastavit, aby došlo k pozastavení databáze do příštího měsíce nebo pokračovat v používání databáze za poplatky účtované podle nastaveného plánu nad bezplatný limit.

Opakem bezserverových výpočetních prostředků jsou zajištěné neboli předem přidělené výpočetní prostředky, které jsou fakturovány po hodině na základě nakonfigurovaných virtuálních jader. Během konfigurace úrovně služby a výpočetních prostředků je možné i povolit automatické pozastavení databáze, pokud je neaktivní po zadanou dobu. Tím se šetří za náklady výpočetních prostředků a databáze se při další aktivitě opět automaticky rozběhne. Uživatel si také volí, zda chce udělat zónu databáze redundantní, a tím chránit svá data před výpadky a případnou ztrátou. Pro redundanci uložení zálohování si volí mezi oblastmi jako místní, zónové nebo geografické. Podle typu oblasti zálohování, možností redundance, nákupním modelem a spotřebovaných gigabajtů uložení za měsíc se určuje i cena zálohování.

Pokud je používáno více databází s proměnlivou zátěží, lze využít funkci Elastický fond SQL pro optimalizaci nákladů a správu. Elastický fond v Azure SQL Database umožňuje sdílet a spravovat výpočetní a úložný prostor mezi sadou databází v rámci jednoho serveru. [10]

Aby mohly služby a prostředky Azure přistupovat k tomuto serveru, je potřeba v konfiguraci přístupu k síti jako metodu připojení povolit veřejný koncový bod a následně pro pravidla brány firewall nastavit „Ano“ pro umožnění službám a prostředkům Azure přistupovat k tomuto serveru a „Ano“ pro přidání aktuální IP adresy klienta pro ruční práci s databází. Tato pravidla lze ale kdykoliv později měnit v přehledu serveru v sekci „Zabezpečení -> Sítě“. Pro více bezpečný přístup k databází je ale vhodnější nakonfigurovat virtuální síť s privátními koncovými body. V části zabezpečení je vhodné povolit identitu serveru, aby bylo možné ověřovat služby Azure bez toho, aby bylo nutné ukládat přihlašovací údaje do kódu aplikace.

6.8.3.2 Vytvoření webového rozhraní API ASP.NET Core

V Microsoft Visual Studiu je potřeba vytvořit nový projekt ze šablony projektů webové rozhraní API ASP.NET Core. V dalším kroku lze vybrat .NET Framework rozhraní .NET 7.0 a konfiguraci nastavit na HTTPS a s používáním kontrolérů pro předpřipravené rozhraní API. Pro snazší testování aplikace z uživatelského rozhraní je možné také povolit podporu OpenAPI (Swagger), kterou lze později před nasazením aplikace z kódu odstranit.

V ASP.NET Core aplikaci je pro lepší přehlednost a údržbu kódu vhodné klíčový soubor aplikace rozdělit na dva – Program.cs a Startup.cs. Program.cs je vstupním bodem aplikace, jehož hlavním účelem je hostování aplikaci a její spouštění, případně provedení základní konfigurace aplikace a inicializaci služeb, které jsou potřeba pro běh aplikace. Startup.cs je soubor, ve kterém je definováno, jak bude aplikace nakonfigurována. Obsahuje metodu „*ConfigureServices()*“, která je použita pro registraci služeb používaných v aplikaci do „*Dependency Injection*“ kontejneru pro získání instance této služby, a metodu „*Configure()*“ pro konfiguraci „*middleware*“ pro zpracování HTTP požadavků, jako jsou např. routování, autentizace, autorizace, logování, atd.

Také standardní rozdělení aplikace na Controllers a Models pro oddělení logiky od datového modelu pomůže udržet aplikaci lépe organizovanou, čitelnou a rozšiřitelnou. V Controllers, které slouží jako most mezi uživatelským rozhraním logikou aplikace, se provádí zpracování HTTP požadavků klientů a příprava dat pro zobrazení. Models obsahují datový model aplikace, který reprezentuje strukturu a chování dat a jejich ukládání do databáze.

6.8.3.3 Komunikace API s databází pomocí SqlConnection

S knihovnou Microsoft.Data.SqlClient lze vytvářet, provádět a spravovat SQL dotazy přímo na databázi. Má vysokou výkonnost a efektivitu, což je důležité pro aplikace, které vyžadují rychlé operace s databází. Pro manipulaci s daty, vytváření připojení k databázi a správu transakcí je ale potřeba napsat více ručního kódu. Lze místo toho využít i ORM knihovnu Entity Framework Core (EF Core), což znamená, že umožňuje mapování dat mezi objekty v kódu a tabulkami v databázi. To zjednodušuje práci s databází tím, že umožňuje pracovat s daty jako s objekty a provádět operace CRUD (Create, Read, Update, Delete) pomocí objektově orientovaného kódu. EF Core se stará o generování SQL dotazů, což eliminuje potřebu psát mnoho ručního kódu pro manipulaci s databází, ale v některých případech může vykazovat mírně nižší výkon než Microsoft.Data.SqlClient, protože zahrnuje další vrstvu mapování dat.

Balíček Microsoft.Data.SqlClient je potřeba doinstalovat v .NET řešení pomocí příkazu „dotnet add package Microsoft.Data.SqlClient“ nebo v okně „Průzkumník řešení“ kliknout na uzel „Závislosti projektu“ a vybrat „Spravovat balíčky NuGet“. Následně vyhledat Microsoft.Data.SqlClient a nainstalovat. Pokud Visual Studio obsahuje pouze offline zdroj balíčku NuGet, který nezahrnuje knihovnu Microsoft.Data.SqlClient, je potřeba v možnostech nastavení v sekci „Správce balíčků NuGet“ přidat do zdroje balíčků další balíček s názvem „nuget.org“ a se zdrojem „<https://api.nuget.org/v3/index.json>“.

Připojení aplikace ke službě Azure SQL Database pomocí připojovacího řetězce bez hesla je bezpečnější, protože zdrojový kód neobsahuje tajné kódy, jako jsou uživatelská jména, hesla nebo přístupové klíče. V přehledu SQL databáze v Azure Portal v sekci nastavení je možné zobrazit připojovací řetězce k vybrané databázi. Zkopírovaný řetězec je potřeba vložit do .NET aplikace do souboru „appsettings.json“ jako „ConnectionStrings“. Následně je potřeba připojovací řetězec bez hesla ze souboru „appsettings.json“ načíst v části Models.

```
"ConnectionStrings": {  
  "AZURE_SQL_CONNECTIONSTRING": "Server=tcp:attendancesystem.database.windows.net,1433;Initial Catalog=GuideTest;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;Authentication=\\\"Active Directory Default\\\";"  
}
```

Zdrojový kód 8: Připojovací řetězec bez hesla pro připojení ASP.NET API k Azure SQL Database

Připojovací řetězec bez hesla nastaví konfigurační hodnotu „*Authentication=“Active Directory Default”*“, která dává knihovně Microsoft.Data.SqlClient pokyn, aby se připojila ke službě Azure SQL Database pomocí třídy s názvem „*DefaultAzureCredential*“. To umožňuje připojení ke službám Azure bez hesla díky knihovně identit Azure, na které závisí klientská knihovna SQL. Třída podporuje více metod ověřování a určuje, které se mají za běhu použít podle různých prostředí. Například pokud aplikace běží místně, ověřuje se prostřednictvím uživatele, který je přihlášen k sadě Visual Studio. U aplikace nasazené do Azure, ten stejný kód vyhledá a použije spravovanou identitu přidruženou k hostované aplikaci, která byla nakonfigurována.

```
public User(IOptions<MyConnectionModel> azureString)
{
    _azureString = azureString;
}
[HttpGet("GetUsers")]
public IEnumerable<string> Get()
{
    string connectionString =
        _azureString.Value.AZURE_SQL_CONNECTIONSTRING;
    var rows = new List<string>();

    using (var conn = new SqlConnection(connectionString))
    {
        conn.Open();
        var command = new SqlCommand("SELECT * FROM Users", conn);
        using SqlDataReader reader = command.ExecuteReader();
        if (reader.HasRows)
        {
            while (reader.Read())
            {
                rows.Add($"{reader.GetString(0)}, {reader.GetString(1)},
{reader.GetString(2)}");
            }
            reader.Close();
            conn.Close();
        }
        return rows;
    }
}
```

Zdrojový kód 9: Připojení ke službě Azure SQL Database z API pomocí Microsoft.Data.SqlClient

6.8.3.4 Publikování webového rozhraní API ASP.NET Core do Azure App Service

Azure App Service je platforma jako služba (PaaS), která umožňuje snadné nasazení do cloudu, provoz a škálování webových aplikací a API bez nutnosti spravovat infrastrukturu. Je ideální pro nasazení webových aplikací postavených v různých technologiích, jako ASP.NET, Node.js, Java, PHP a Python, nebo pro nasazení API a služeb, které poskytují přístup k funkcím a datům pro jiné aplikace. Umožňuje automatické škálování podle aktuální zátěže, poskytuje nástroje pro nasazení aplikací z různých zdrojů, jako Git repositářů, Azure DevOps, Docker kontejnerů, ale aplikaci lze nasadit i pomocí Azure Portalu, Azure CLI nebo Visual Studio, a integruje s dalšími Azure službami, jako Azure SQL Database. Azure App Service je k dispozici v různé řadě plánů úrovně Free, Basic, Premium nebo jako služba v izolovaném prostředí. Podle výběru plánu se za výpočetní kapacitu platí po sekundách, a to pouze za využití prostředky.

V Azure App Service je možné vytvořit novou webovou aplikaci jako službu a následně do ní nahrát připravené webové rozhraní nebo lze přímo ve Visual Studiu vytvořit službu webové aplikace a připravenou aplikaci přímo nasadit do Azure v jednom pracovním postupu. Při publikování řešení z Visual Studia se je potřeba v dialogovém okně zvolit „Azure“ a následně službu „Azure App Service (Windows)“ jako cíl nasazení. Během vytvoření nové služby je nutné vyplnit název aplikace a zvolit předplatné, skupinu prostředku a plán hostování. Tím je ve Visual Studiu vytvořen prostředek služby App Service v Azure. Krok nastavení služby „API Management“ je možné přeskocit, nastavení je tímto dokončeno a služba je vytvořena. Následně je do služby Azure přes tlačítko „Publikovat“ aplikace nasazena. Při každé provedené změně v aplikaci probíhá její nasazení už jen pouze přes toto tlačítko. [11]

6.8.3.5 Konfigurace zabezpečeného připojení mezi službou App Service a databází SQL pomocí spravované identity

Jelikož bylo připojení mezi službami App Service a Azure SQL Database vytvořeno bez hesla, je potřeba nastavit spravovanou identitu pro nasazené API ve službě App Service a tuto identitu přiřadit databázi SQL, aby mělo API oprávnění přistupovat k datům z Azure databáze. Spravovanou identitu je možné vytvořit výběrem „Identita“ z levého navigačního panelu u vybrané služby App Service. Na kartě „Přiřazeno systémem“ je potřeba, aby byl stav nastavený na „Zapnuto“. Tím je vytvořena spravovaná identita přiřazená systémem, která je pojmenovaná stejně jako název vytvořené služby. Takovéto identity jsou svázané s životním cyklem prostředku a jsou ukončeny spolu s odstraněním aplikace.

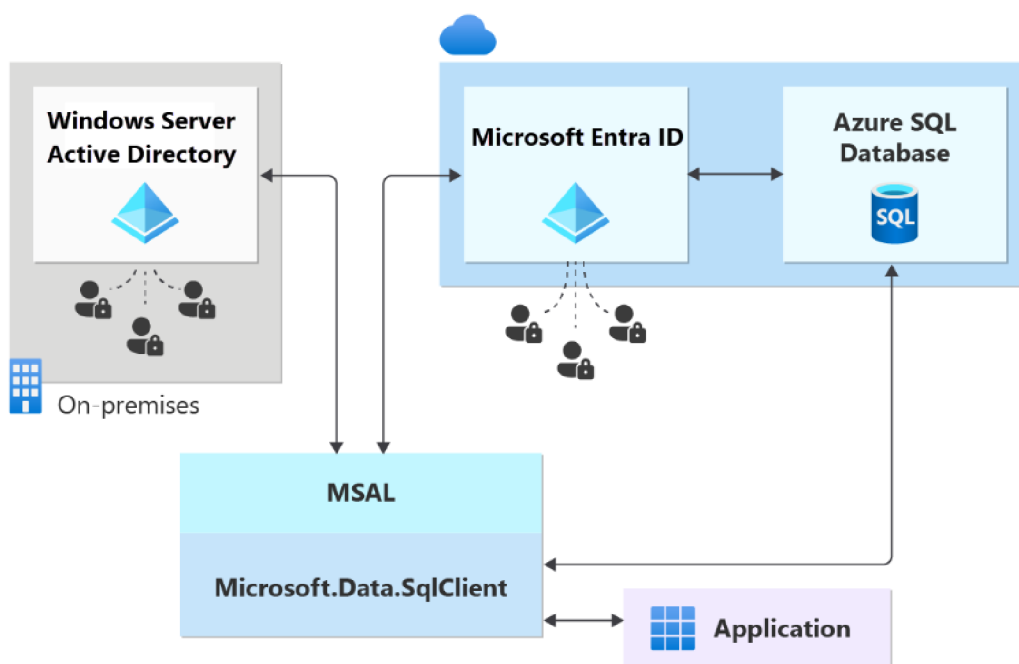
V databázi je potřeba vytvořit uživatele aplikace a přiřadit mu potřebné oprávnění pro čtení, zapisování a upravování dat a schématu databáze. Ve zvolené databázi je možné spravovat dotazy v části „Editor dotazů (Preview)“. Přihlášení může proběhnout pomocí Microsoft Entra ID nebo účtu správce databáze s povolenou místní IP adresou v pravidlech brány firewall v části nastavení zabezpečení a sítě serveru databáze. V editoru je možné vytvářet příkazy pro databázi pomocí jazyka Transact-SQL (T-SQL). Následujícím skriptem je tak vytvořen uživatel databáze, který se odkazuje na spravovanou identitu instance služby App Service, a přiřazuje mu potřebné role. Pro bezpečnější přístup je vhodné nakonfigurovat více identit s přesně definovanými oprávněními pro různé úlohy, aby veškeré operace neprobíhaly pouze pomocí jedné spravované identity.

```
CREATE USER attentancesystemapp FROM EXTERNAL PROVIDER;  
ALTER ROLE db_datareader ADD MEMBER attentancesystemapp;  
ALTER ROLE db_datawriter ADD MEMBER attentancesystemapp;  
ALTER ROLE db_ddladmin ADD MEMBER attentancesystemapp;  
GO
```

Zdrojový kód10: T-SQL příkaz pro vytvoření uživatele databáze mapovaného na spravovanou identitu aplikace a přiřazení rolí

Díky vytvoření instance služby App Service ke službě Azure SQL Database si knihovna Microsoft.Data.SqlClient, která je součástí API aplikace, automaticky zjistí spravovanou identitu stejně jako místního uživatele sady Visual Studio a ke službě se připojí. Otevřením URL adresy aplikace doplněnou o „/swagger/index.html“ je možné spustit testovací stránku.

Ověření uživatelů a jejich přístupu k zabezpečeným webovým rozhraním API umožňuje knihovna Microsoft Authentication Library (MSAL). Ta může získávat tokeny zabezpečení z Microsoft Identity Platform, díky čemuž není nutné používat knihovny OAuth přímo v kódu aplikace. [12]



Obrázek 22: Zabezpečená komunikace mezi službou App Service a SQL databází přes Microsoft.Data.SqlClient díky automatickému ověřování přes Microsoft Entra ID

6.8.3.6 Registrování aplikace v Microsoft Entra ID

Registrace nové aplikace v Microsoft Entra ID je mostem pro komunikaci mezi SPFx a službou App Service, jelikož umožňuje SPFx získat přístupový token potřebný pro autorizaci při volání API v Azure. Tím je zajištěno, že pouze oprávněné aplikace mají přístup k chráněným zdrojům. Díky tomu je také možné spravovat udělování přístupových práv k určitým API. Registrovaná aplikace umožňuje sledovat a auditovat, kdo a kdy aplikaci používá, a zobrazit protokol přístupových požadavků a udělených tokenů. Také zahrnuje implementaci dalších bezpečnostních opatření, jako je vícefaktorové ověřování, certifikáty nebo správa přístupu na základě rolí.

V Microsoft Entra ID je v levém navigačním panelu potřeba zvolit „Registrace aplikací“ a následně „Nová registrace“, kde se pouze vyplní název nové aplikace. Tím je automaticky vytvořeno ID aplikace (klienta) se kterým se nadále pracuje. V registrované aplikaci v části „Zpřístupnit rozhraní API“ je potřeba vytvořit identifikátor URI pro ID aplikace a přidat obor „*access_as_user*“ pro přístup k datům a funkcím aplikace chráněné tímto rozhraním API. Díky tomu aplikace, která přistupuje ke službám Azure přes ID této registrované aplikace, bude mít jen takové přístupy, které jsou v rozhraní API definované. [18]

Ověřovací knihovna platformy Microsoft Identity získává platný přístupový token pomocí principu služby „*SharePoint Online Client Extensibility*“. Ta je k dispozici v Microsoft Entra ID mezi registrovanými aplikacemi. Pro ověření a autorizaci je potřeba v této službě v části „Oprávnění rozhraní API“ přidat nové oprávnění odkazující na vytvořený obor definovaný rozhraním API nově registrované aplikace a udělit souhlas správce.

6.8.3.7 Nastavení CORS pro přístup ke zdrojům v jiné doméně

Veškeré požadavky, které jsou z SPFx aplikace prostřednictvím prohlížeče odeslány do API hostované v Azure, jsou blokovány. Toto bezpečnostní omezení webových prohlížečů „*same-origin policy*“ brání webové stránce volání rozhraní API v jiné doméně. Jelikož SPFx aplikace hostovaná v SharePoint neodpovídá původní doméně API v App Service v Azure, jsou ve výchozím nastavení tyto požadavky blokovány.

Služba App Service v Azure podporuje CORS (Cross-Origin Resource Sharing), což je HTTP funkce, která bezpečným způsobem umožňuje aplikaci běžící v jedné doméně přistupovat ke zdrojům v jiné doméně. Vložením URL adresy aplikace v jiné doméně se povolí sdílení prostředků mezi zdroji, což umožňuje JavaScript kódu spuštěném v prohlížeči na externím hostiteli spolupracovat s backendem. Pokud by aplikace vyžadovala i odesílání přihlašovacích údajů (cookies soubory, ověřovací tokeny), byla by vyžadována i hlavička odpovědi „*Access-Control-Allow-Credentials*“, kterou je také možné ve službě App Service v části „CORS“ povolit – nastavit její hodnotu na „*true*“.

Požadavek se nejprve dotazuje na omezení CORS odesláním předletového požadavku „*Options*“, který obsahuje záhlaví, metodu a původní doménu. Na základě nakonfigurovaných pravidel CORS služba vyhodnocuje, které původní domény, metody a hlavičky požadavků mohou být specifikovány ve skutečném požadavku. Pokud je pro službu povolen CORS a existuje pravidlo CORS, které odpovídá požadavku, služba odpoví stavovým kódem 200 (OK) a zahrne do odpovědi požadované hlavičky „*Access-Control*“. Pokud je požadavek na kontrolu před výstupem přijat a odpověď je vrácena, prohlížeč odešle skutečný požadavek na API. Pokud pro službu není povolen CORS nebo žádné pravidlo CORS neodpovídá požadavku před výstupem, služba odpoví stavovým kódem 403 (Zakázáno) a záhlaví CORS „*Access-Control*“ se nevrátí. Prohlížeč také odmítne skutečný požadavek okamžitě, pokud je žádost o kontrolu pravidel CORS před výstupem zamítnuta nebo pokud kontrolní požadavek neobsahuje požadované záhlaví CORS. [15]

6.8.3.8 Komunikace SPFx s webovým rozhraním pomocí AadHttpClient

SPFx se připojuje k API zabezpečenému pomocí Microsoft Entra ID, proto je aplikaci potřeba udělit potřebná oprávnění k připojení. Pomocí AadHttpClient je možné se jednoduše připojit k rozhraním API zabezpečeným pomocí Microsoft Entra ID, aniž by bylo nutné implementovat tok OAuth.

Obvykle, aby se vlastní podnikové aplikace Microsoft 365 vytvořené organizací mohly připojit ke zdrojům, musí získat platný přístupový token k určitému zdroji, což je součástí procesu autorizace OAuth. Jenže aplikace na straně klienta, které nejsou schopny ukládat tajné informace, jako je SPFx řešení, používají pro implementaci autorizace specifický typ toku zvaný implicitní tok OAuth. V SPFx je ale tento proces již řešený pomocí MSGraphClient a AadHttpClient. Aplikace může požádat o přístup ke konkrétním prostředkům v manifestu řešení a při nasazení do katalogu aplikací v SharePointu je vyzván správce, aby požadovaná oprávnění spravoval. Nebo jsou všechna potřebná oprávnění už udělena v celém tenantovi v Microsoft Entity ID do aplikace „*SharePoint Online Client Extensibility*“, kterou používá SPFx v toku OAuth pro poskytování řešení s platnými přístupovými tokeny.

V SPFx řešení je potřeba importovat AadHttpClient a následně vytvořit novou instanci a předat zdroj, ke kterému je možné se připojit. AadHttpClient je propojen s konkrétním prostředkem, proto je potřeba pro každý prostředek k připojení vytvořit novou instanci. V tomto případě je zdrojem ID klienta aplikace registrované v Microsoft Entra ID, která byla vytvořena dříve a používá se k zabezpečení podnikového rozhraní API. Na tuto aplikaci je odkázáno v části „Oprávnění rozhraní API“ v hlavním objektu „*SharePoint Online Client Extensibility*“, díky kterému je získán platný přístupový token. ID klienta je předáno jako druhý parametr v konstruktoru AadHttpClient jako GUID. Následně už je možné zadat webový požadavek pro komunikaci s API v Azure App Service. S tímto ID klienta registrované aplikace lze pracovat pouze v rámci stejného tenanta. Pro udělení oprávnění pro aplikaci definovanou v jiném tenantovi, je potřeba nejprve přidat instanční objekt pro danou aplikaci ve své klientovi. [14]

Defaultně nemá SPFx řešení žádný přístup k podnikovým rozhraním API. Tento přístup je možné povolit ručně v Microsoft Entry ID nebo je nutné zadat žádost o povolení přímo z SPFx projektu ze souboru „*config/package-solution.json*“. Vytvoření nové vlastnosti s názvem „*webApiPermissionRequests*“ odkazuje na vytvořenou aplikaci registrovanou v Microsoft Entra ID, která se používá k zabezpečení rozhraní API. Hodnota „*resource*“ odkazuje na název registrované aplikace a hodnota „*scope*“ určuje rozsah oprávnění. [8]

```

import { AadHttpClient, HttpClientResponse } from '@microsoft/sp-http';

private async loadUsers(): Promise<void>{
    this.props.webpartcontext.aadHttpClientFactory.getClient('api://xxxxxxxx
x-xxxx-xxxx-xxxx-xxxxxxxxxxxxx')
    .then((client: AadHttpClient): void => {
        client.get('https://attendancesystemapp.azurewebsites.net/api/User/Us
ers', AadHttpClient.configurations.v1)
        .then((response: HttpClientResponse): Promise<any[]> => {
            return response.json();
        })
        .then((users: string[]): void => {
            console.log(users)
        })
        .catch((e: Error) => console.log(e));
    })
    .catch((e: Error) => console.log(e));
}

```

Zdrojový kód 10: Připojení SPFx k rozhraní API v App Service pomocí AadHttpClient

6.8.3.9 Nastavení zabezpečeného přístupu k API pomocí API Scope

Jelikož do webového rozhraní API ASP.NET nebyla přidána konfigurace pro autentizaci a autorizaci pomocí služby Azure AD, je hostovaná aplikace veřejně přístupná, takže uživatelé mohou přistupovat k jejím endpointům nebo službám bez nutnosti jakékoli autentizace. To může být vhodné pro veřejné webové stránky nebo veřejné API, ale pro aplikace, které pracují s citlivými daty nebo vyžadují ověření uživatele, je důležité implementovat autentizaci a autorizaci pro ochranu těchto zdrojů před neoprávněným přístupem. Proto je potřeba do souboru „*appsettings.json*“ v ASP.NET přidat vlastnost Azure AD a následně autentizovat uživatele pomocí protokolu OpenID Connect a tím poskytnout zabezpečený přístup k API pouze přihlášeným uživatelům nebo povoleným SPFx řešením. [13]

```

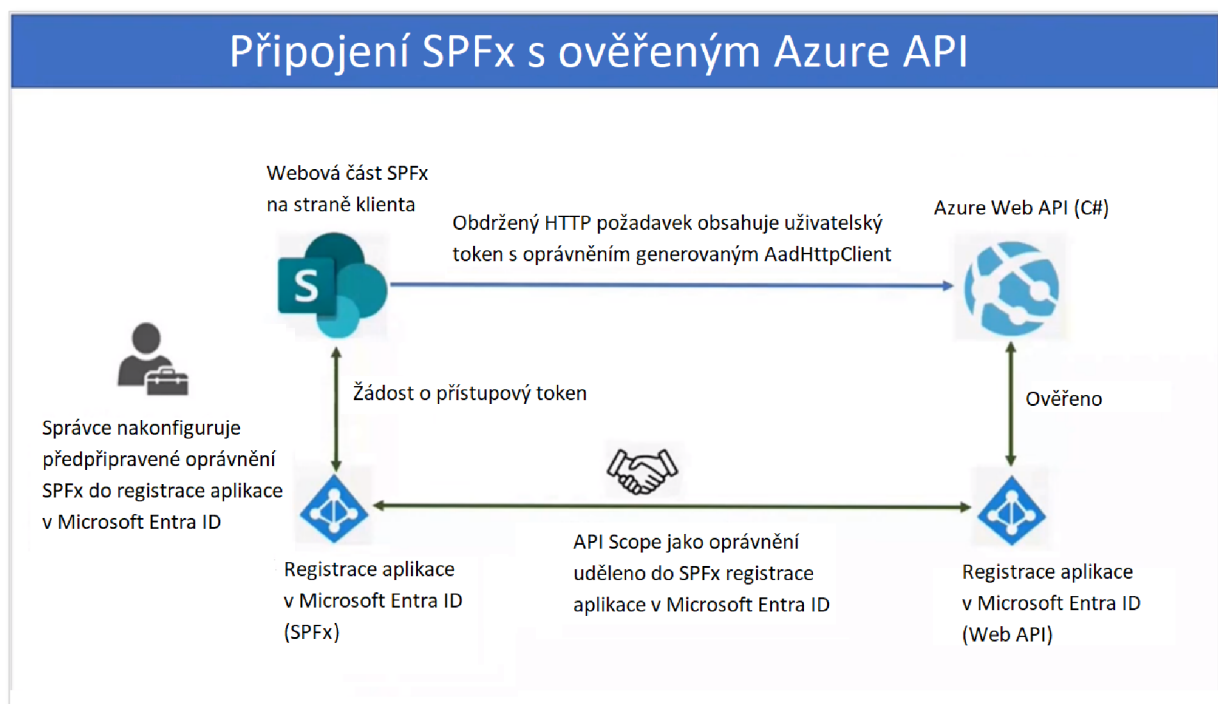
"AzureAd": {
    "Instance": "https://login.microsoftonline.com/",
    "Domain": "xxxxxxxx.onmicrosoft.com",
    "TenantId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx",
    "ClientId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx",
    "CallbackPath": "/signin-oidc",
    "Scopes": "access_as_user"
},

```

Zdrojový kód 11: Zabezpečený přístup k API pouze přihlášeným uživatelům nebo SPFx aplikaci

6.8.3.10 Průběh zabezpečené komunikace mezi klientskou částí SPFx a API ve službě Azure App Service

Konfigurací přístupových práv pro SPFx aplikaci v registrované aplikaci „SharePoint Online Client Extensibility“ se určuje, k jakým zdrojům v Microsoft a Azure bude mít aplikace přístup. Zaregistrováním další aplikace pro Web API v Microsoft Entra ID se získá identifikátor klienta aplikace (Client ID) a další informace, které jsou následně použity pro ověření při komunikaci s Azure Web API. V této registrované aplikaci je potřeba nastavit další oprávnění, které určují, jaké činnosti může SPFx aplikace provádět v rámci Web API. Nakonec je potřeba tuto registrovanou aplikaci přiřadit mezi oprávnění „SharePoint Online Client Extensibility“ aplikace, což zajišťuje, že SPFx aplikace může komunikovat s Web API hostovanou v Azure a provádět požadované operace.



Obrázek 23: Proces propojení webové části na straně klienta SPFx s autentizovaným Azure API

Když jsou SPFx aplikace a Azure Web API úspěšně propojeny a správně nakonfigurovány pomocí Microsoft Entra ID registrace aplikací, probíhá komunikace mezi nimi následovně:

- Po přihlášení uživatele do SPFx, webová část na straně klienta požádá o přístupový token od Microsoft Entra ID, který získá na základě autentizace uživatele.
- Když pak uživatel v SPFx aplikaci vyvolá akci, která vyžaduje data z Azure Web API, SPFx sestaví HTTP požadavek obsahující ID klienta registrované aplikace pro Web API, cílovou URL adresu API a požadované parametry.
- SPFx ale nejprve požádá o přístup k Azure Web API pomocí získaného přístupového tokenu, který obsahuje informace o oprávněních.
- Registrovaná aplikace Azure pro Web API je nakonfigurována tak, aby autentizovala a autorizovala požadavky od aplikace SPFx a zároveň komunikovala s rozhraním API. Na základě přidělených oprávnění v této registrované aplikaci je definován rozsah přístupu (API Scope), který má aplikace SPFx při interakci s Azure API. Proto tato aplikace ověří přístupový token, zda SPFx aplikace má oprávnění k volání daného API.
- Pokud SPFx dostane potvrzení o oprávnění volání API, zašle HTTP požadavek, do jehož hlavičky přidá získaný přístupový token.
- Tímto způsobem Web API pozná, že požadavek pochází již od ověřené SPFx aplikace, žádost zpracuje a odešle zpět odpověď obsahující požadovaná data.
- Díky tomuto postupu může klientská část webového prvku SPFx bezpečně komunikovat s Azure Web API, což zajišťuje, že k chráněným prostředkům mohou přistupovat pouze autorizovaní uživatelé nebo aplikace.

7 Testování docházky

Aplikace docházkového systému splňuje všechny stanovené funkční požadavky a její požadované funkcionality byly úspěšně implementovány a fungují tak, jak bylo plánováno. Záložka nastavení ale ještě postrádá některé funkce pro plnohodnotné fungování systému, jako je úpravu již existujících dat o uživateli z databáze týkající se úprav úvazků, dovolených nebo přesčasů. Tyto funkce bude potřeba doimplementovat stejně tak jako zobrazování případných zaznamenaných změn v docházce. Uživatelé pozitivně hodnotí uživatelské rozhraní pro svou plynulou použitelnost, jednoduchost a intuitivnost. Jsou spokojeni s rychlostí odezvy a efektivitou aplikace při plnění úkolů.

U záložky pro samotný záznam docházky kladně hodnotí, že plní všechny funkce původního softwaru pro záznam docházky, ale je snazší na používání a celkově přehlednější s prvky typickými pro Teams a jednodušším designem. Neměli problém se v aplikaci zorientovat. Oceňují funkci nastavení více přestávek k jednomu záznamu, což v dosavadním systému nebylo možné. To přidává na přesnosti denního rozložení hodin. Ovšem nutnost nejprve uložit zapsanou přestávku do seznamu a pak až aktualizovat záznam, aby byla přestávka následně přidána k záznamu, je nezvyklé, proto občas tuto změnu zapomenou potvrdit.

Uživatelé si chválí funkci založení/ukončení denní činnosti s aktuálním časem, která je pro ně přínosem například pro rychlý záznam při výjezdech nebo pro trasování cest. To podchytává chybějící funkci v aktuálním Excelu. Líbí se jim možnost srovnávání záznamů dle datumu při zpětném doplňování docházky a také možnost přidání práce na jiném projektu ve stejný den.

Ocenili by ale ještě funkci, která by zobrazovala celkový počet již odpracovaných hodin za aktuální nebo konkrétně zvolený den pro získání větší kontroly nad svou pracovní dobou během dne, stejně tak jako zvýraznění tučným písmem celkového počtu odpracovaných hodin v tabulce u každého záznamu. Při větším obsahu záznamů v tabulce a potřebě posunout stránku dolů by bylo vhodné, aby záhlaví tabulky zůstalo ukotvené. Vybraný záznam by pro lepší přehled mohl být v tabulce zvýrazněný. Funkce tlačítka „Complete“ musela být uživatelům nejprve vysvětlena.

Tabulka pro zobrazení statistik přehledně vychází z dříve zavedeného softwaru pro vedení docházky, tedy zobrazuje ty základní údaje, které uživatel očekává. Uživatel by ale také uvítal ještě další statistiky, jako jsou celkové počty odpracovaných hodin na projektech za zvolený měsíc nebo za celý rok.

Dalšími možnými kroky v budoucnu může být vylepšení docházkového systému o další funkce, jako je filtrování a zobrazení dalších přehledů a statistik, jako jsou celkové součty odpracovaných hodin uživatele na jednotlivých projektech během měsíce či náhledy celkového počtu odpracovaných hodin na projektech přímo v docházkovém systému ještě před generováním dokumentů. Aplikace by mohla také zobrazovat aktuální počet odpracovaných hodin během dne nebo obsahovat další záložku, která by fungovala jako formulář pro vyplnění cestovních náhrad nebo formulář servisu.

Docházkový systém je aktuálně připraven na míru v rámci jednoho tenantu, proto dalšími kroky mohou být úpravy, které umožňují implementaci i do jiných tenantů. To zahrnuje i v části nastavení aplikace možnost výběru si konkrétních zdrojů, se kterými bude aplikace pracovat, jako jsou projekty a typ práce, které nemusí být importovány pouze ze seznamů SharePointu, ale také např. z Excelu, aplikace MS Project, jiné databáze nebo mohou být zadány ručně.

Jelikož je SPFx aplikace hostovaná na SharePointu a její obsah je pouze zobrazen na platformě Teams, je možné s ní pracovat z jakéhokoliv zařízení, ve kterém má uživatel přístup k Teams, tedy i z mobilního zařízení. Díky tomu je práce s docházkou efektivnější, jelikož většina uživatelů si může příchod nebo odchod do práce a případné pauzy zaznamenávat rychle přímo z mobilu. Ovšem pro lepší přehled zobrazené docházky v mobilu je potřeba změnit zobrazení ovládacích prvků, jelikož se pracuje se zařízením s jiným rozlišením.

Pro zdrojové kódy k této diplomové práci je možné nahlédnout do git repositáře na adrese: <https://github.com/KristynaFrydrychova/AttendanceSystem> (14. 5. 2024)

Závěr

Cílem diplomové práce bylo prozkoumat možnosti tvorby a integrace aplikací do prostředí Microsoft Teams, zanalyzovat existující softwarové řešení a procesy související s docházkou, specifikovat funkční požadavky, vytvořit návrh vzorové aplikace a implementovat pokročilý docházkový systém pro malé a střední podniky. Docházkový systém byl zvolen z důvodu, že v sadě aplikací balíčku Microsoft 365 chybí služba, která by zaměstnancům umožňovala vést pokročilejší zaznamenávání jejich docházky a přiřazování vykázaných hodin k projektům.

Nejprve jsem se zabývala interními aplikacemi potřebnými pro řízení podniku a vypsalala jejich funkční požadavky. Následně jsem prozkoumala možnosti rozšiřitelnosti platformy Teams a vyzdvihla především tvorbu a integraci osobních aplikací, jako jsou Power Apps, SPFx nebo existující webové aplikace, které jsem mezi sebou porovnávala a popsala jejich limity.

Následně jsem provedla analýzu již existujících softwarových řešení pro vedení docházky zaměstnanců a sepsala funkční požadavky pro výsledný docházkový systém. Díky návrhu interaktivního prototypu uživatelského rozhraní jsem získala zpětnou vazbu od potenciálních uživatelů.

Na základě předchozí analýzy a rešerši způsobu tvorby aplikací do Microsoft Teams jsem zvolila způsob implementace vzorové aplikace pomocí řešení SPFx, tedy frameworku pro rozšiřování SharePoint webu. Výběr vhodného technického řešení a metody integrace do prostředí Microsoft Teams jsem uskutečnila s ohledem na potřeby a požadavky cílového uživatelského segmentu.

Provedla jsem podrobný rozbor implementace aplikací ve zvoleném technickém řešení, což zahrnovalo přípravu prostředí pro tvorbu SPFx aplikací a následné vytvoření webové části. Součástí bylo také vytvoření manifest souboru, provedení konfigurací, nahrání na SharePoint a testování aplikace. Podrobný rozbor jsem také věnovala tomu, jaké způsoby využívá SPFx pro práci s daty a jejich zabezpečenému přenosu napříč aplikacemi.

Na závěr jsem ve vybraném řešení implementovala docházkový systém jako vzorovou aplikaci a popsala ji z hlediska funkčního řešení. Provedené testy potvrdily funkčnost a spolehlivost navrženého systému. Vyhodnocení výsledků testů poskytlo cenné poznatky pro další vývoj a optimalizaci aplikace. Navržený docházkový systém se ukázal jako efektivní nástroj pro správu pracovního času a zajištění transparentního monitorování docházky zaměstnanců pro malé a střední podniky.

Citovaná literatura

1. **365 Software LLP. 2020.** TimeTracker. *Microsoft AppSource*. [Online] 9. listopadu 2020. [Citace: 4. dubna 2023.] <https://appsource.microsoft.com/en-us/product/office/wa200002330?tab=overview&exp=kyyw>.
2. **Al Rafay Consulting. 2021.** PowerAPPs Limitation. *Al Rafay Consulting*. [Online] 12. prosince 2021. [Citace: 6. března 2023.] <https://www.alrafayglobal.com/power-apps/powerapps-limitation-with-sharepoint-online-vs-sharepoint-on-premises-configuring-data-gateway-for-sharepoint-on-premises/>.
3. **Andaloussi, Mehdi Slaoui. 2019.** PowerApps Optimization Techniques. *Microsoft Power Apps*. [Online] 21. srpna 2019. [Citace: 14. dubna 2023.] <https://powerapps.microsoft.com/fr-ch/blog/powerapps-optimization-techniques/>.
4. **AttendanceBot. 2020.** Top 8 Microsoft Teams Apps. *AttendanceBot Blog*. [Online] 10. března 2020. [Citace: 4. dubna 2023.] <https://www.attendancebot.com/blog/top-microsoft-teams-apps/>.
5. **DO365.** Docházka Lokace. *DO365*. [Online] [Citace: 4. dubna 2023.] <https://do365.cloud/produkt/aplikace-dochazka-lokace/>.
6. **GIRITON.** Snadná evidence docházky zaměstnanců. *GIRITON*. [Online] [Citace: 4. dubna 2023.] <https://giriton.com/cs>.
7. **IT MUNI.** Microsoft Outlook. *IT MUNI*. [Online] [Citace: 29. března 2023.] <https://it.muni.cz/sluzby/microsoft-outlook>.
8. **Kalaiselvi, Selvi. 2019.** Creating SharePoint Framework (SPFx) Connections to Azure APIs. *ais*. [Online] 18. prosince 2019. [Citace: 10. října 2023.] <https://www.ais.com/creating-sharepoint-framework-spfx-connections-to-azure-apis/>.
9. **Kamal, Heba. 2023.** How To Debug A PowerApps App? *Power Platform Geeks*. [Online] 15. března 2023. [Citace: 17. dubna 2023.] <https://devworx.net/how-to-debug-a-powerapps-app/>.
10. **Microsoft. 2023.** Automated backups in Azure SQL Database. *Microsoft Learn*. [Online] 27. listopadu 2023. [Citace: 14. března 2024.] <https://learn.microsoft.com/en-us/azure/azure-sql/database/automated-backups-overview?view=azuresql>.
11. —. **2023.** Azure App Service on Windows pricing. *Microsoft Azure*. [Online] 31. října 2023. [Citace: 9. listopadu 2023.] <https://azure.microsoft.com/en-us/pricing/details/app-service/windows/>.
12. —. **2023.** Connect to and query Azure SQL Database using .NET and the Microsoft.Data.SqlClient library. *Microsoft Learn*. [Online] 11. července 2023. [Citace: 22. října 2023.] <https://learn.microsoft.com/cs-cz/azure/azure-sql/database/azure-sql-dotnet-quickstart?view=azuresql&tabs=visual-studio%2Cpasswordless%2Cazure-portal%2Cportal>.

13. —. **2024.** Connect to Azure AD-secured APIs in SharePoint Framework solutions. *Microsoft Learn*. [Online] 11. ledna 2024. [Citace: 8. dubna 2024.] <https://learn.microsoft.com/en-us/sharepoint/dev/spfx/use-aadhttpclient>.
14. —. **2022.** Connect to SharePoint APIs. *Microsoft Learn*. [Online] 29. června 2022. [Citace: 24. února 2024.] <https://learn.microsoft.com/en-us/sharepoint/dev/spfx/connect-to-sharepoint>.
15. —. **2023.** Cross-Origin Resource Sharing (CORS) support for Azure Storage. *Microsoft Learn*. [Online] 11. července 2023. [Citace: 21. října 2023.] <https://learn.microsoft.com/en-us/rest/api/storageservices/cross-origin-resource-sharing--cors--support-for-the-azure-storage-services?source=recommendations>.
16. —. **2022.** Map your use cases to Teams app features. *Microsoft Learn*. [Online] 9. prosince 2022. [Citace: 23. února 2023.] <https://learn.microsoft.com/en-us/microsoftteams/platform/concepts/design/map-use-cases>.
17. —. **2023.** Overview of the SharePoint Framework. *Microsoft Learn*. [Online] 21. listopadu 2023. [Citace: 17. března 2024.] <https://learn.microsoft.com/en-us/sharepoint/dev/spfx/sharepoint-framework-overview?source=recommendations>.
18. —. **2023.** Use Microsoft Entra authentication. *Microsoft Learn*. [Online] 30. září 2023. [Citace: 22. října 2023.] <https://learn.microsoft.com/en-us/azure/azure-sql/database/authentication-aad-overview?view=azuresql>.
19. —. **2022.** Why the SharePoint Framework? *Learn*. [Online] 29. června 2022. [Citace: 17. března 2024.] <https://learn.microsoft.com/en-us/sharepoint/dev/spfx/sharepoint-framework-why>.
20. **Schouten, Robert. 2020.** Creating a Microsoft Teams app: how easy is it? *ROBERT SCHOUTEN*. [Online] 14. ledna 2020. [Citace: 21. listopadu 2022.] <https://robertschouten.com/2020/01/14/creating-a-microsoft-teams-app-how-easy-is-it/>.
21. **Sudharsan, Kesavanarayanan. 2020.** SPFx – Using MSGraph API. *KNOWLEDGE SHARE*. [Online] 22. června 2020. [Citace: 10. října 2023.] <https://spkknowledge.com/2020/06/22/spfx-using-msgraph-api/>.
22. **Systemart s.r.o.** Jednoduchá evidence. *Výkaz práce*. [Online] [Citace: 4. dubna 2023.] <https://www.vykazprace.cz/>.
23. **Teamogy.** Teamogy Systém. *Teamogy*. [Online] [Citace: 4. dubna 2023.] <https://www.teamogy.com/cs/system/>.
24. **Tech Peanuts. 2021.** How to run any website as a Teams app! *Tech Peanuts*. [Online] 7. ledna 2021. [Citace: 26. února 2023.] <https://tech-peanuts.com/2021/01/07/how-to-run-any-website-as-a-teams-app/>.
25. **Timeneye.** The time tracking tool that your Microsoft Suite is missing. *Timeneyey*. [Online] [Citace: 4. dubna 2023.] <https://www.timeneye.com/>.

Příloha A

Přehled kardinality a modality mezi entitami v databázi (kapitola 5.1)

Tabulka 1	Tabulka 2	Kardinalita	Modalita
RECORD	USER	N:1	optional–mandatory
Pro každý záznam v tabulce RECORD musí existovat vždy a pouze jeden uživatel z tabulky USER a každý uživatel může existovat ve více záznamech, ale také nemusí mít žádný.			
RECORD	PROJECT	N:1	optional–optional
Pro každý záznam v tabulce RECORD může, ale nemusí existovat maximálně jeden projekt z tabulky PROJECT, a projekt může existovat ve více záznamech, ale také v žádném.			
RECORD	WORKTYPE	N:1	optional–optional
Pro každý záznam v tabulce RECORD může, ale nemusí existovat maximálně jeden typ práce z tabulky WORKTYPE, a každý typ práce může existovat ve více záznamech, ale také nemusí být v žádném.			
PAUSE	RECORD	N:1	optional–mandatory
Pro každý záznam v tabulce PAUSE musí vždy existovat jedno ID přiřazeného záznamu z tabulky RECORD a k jednomu záznamu v tabulce RECORD nemusí existovat žádná přestávka, ale může jich být i více.			
CHANGES_R	RECORD	N:1	optional–mandatory
Pro každou změnu v tabulce CHANGES_R je vždy přiřazen pouze jeden záznam z tabulky RECORD a jeden záznam z tabulky RECORD obsahuje více změn nebo žádnou.			
CHANGES_R	USER	N:1	optional–mandatory
Každá změna z tabulky CHANGES_R musí mít vždy přiřazeného pouze jednoho uživatele z tabulky USER a jeden uživatel může provést několik změn, ale také žádnou.			
CHANGES_P	PAUSE	N:1	optional–mandatory
Pro každou změnu v tabulce CHANGES_P je vždy přiřazen pouze jeden záznam z tabulky PAUSE a jeden záznam z tabulky PAUSE obsahuje více změn nebo žádnou.			
CHANGES_P	USER	N:1	optional–mandatory
Každá změna z tabulky CHANGES_P musí mít vždy přiřazeného pouze jednoho uživatele z tabulky USER a jeden uživatel může provést několik změn, ale také žádnou.			

WORKINGHOURS	USER	N:1	mandatory–mandatory
Pro každý záznam v tabulce WORKINGHOURS existuje přiřazený vždy jeden uživatel z tabulky USER a jeden uživatel musí mít vždy nastavený alespoň jeden úvazek, ale může existovat i více záznamů v případě, že vznikl nový typ úvazku.			
VACATION	USER	N:1	mandatory–mandatory
Pro jeden záznam v tabulce VACATION musí být vždy přiřazen jeden uživatel z tabulky USER a jeden uživatel může mít více záznamů v tabulce VACATION, ale vždy alespoň jeden.			
DIFF	USER	N:1	optional–mandatory
Pro každý záznam v tabulce DIFF je nastaven vždy jeden uživatel z tabulky USER a uživatel má může mít jeden nebo více záznamů v tabulce DIFF, ale také nemusí mít žádný.			
OVERTIME	USER	N:1	optional–mandatory
Pro každý záznam v tabulce OVERTIME musí být vždy přiřazen pouze jeden uživatel z tabulky USER a záznam v tabulce USER nemusí mít žádný záznam v tabulce OVERTIME, ale může jich obsahovat i více.			
TOTAL	USER	N:1	optional–mandatory
Každá záznam v tabulce TOTAL má vždy jednoho uživatele z tabulky USER, ale USER může být přiřazený k více hodnotám v tabulce TOTAL, ale také k žádné.			

Příloha B

Vytvoření aplikace pomocí SPFx a způsob její integrace do prostředí Microsoft Teams (kapitola 6)

1. Příprava prostředí (kapitola 6.2):

- Stáhnout a nainstalovat všechny potřebné nástroje (VS Code, Node.js verze 14, v16 nebo v18, SharePoint Online Management Shell)
- Přidat a nastavit „AppCatalog“ pro přidávání aplikací na SharePoint pomocí balíčku „sppkg“:
 - Otevřít SharePoint jako správce -> Další funkce -> Aplikace -> Katalog aplikací
- Povolit uživatelům přidávat vlastní aplikace v Teams:
 - Otevřít Teams jako správce -> Aplikace Teams -> Zásady oprávnění -> Globální -> Vlastní aplikace -> Povolit všechny aplikace (nebo jen konkrétní)
- Povolit CDN pro ukládání statických prostředků do mezipaměti k prohlížečům:
 - Přihlásit se jako tenant přes SharePoint Online Management Shell:
`Connect-SPOService -Url https://{název_tenantu}-admin.sharepoint.com`
 - `Set-SPOTenantCdnEnabled -CdnType Public`
 - `Add-SPOTenantCdnOrigin -CdnType Public -OriginUrl */cdn`
 - Zkontrolovat přidání: `Get-SPOTenantCdnOrigins -CdnType Public`
- Nainstalovat globálně Yeoman a Gulp v příkazovém řádku:
 - `npm install -g yo gulp` případně za příkaz přidat `--legacy-peer-deps` pro možnost používat starší kompatibilní verze nebo `--force` pro vynucenou instalaci balíčků i přes jejich případnou nekompatibilitu
 - `npm install gulp yo @microsoft/generator-sharepoint --global`

2. Vytvoření základu SPFx aplikace spolu s FluentUI (kapitola 6.3):

- V příkazovém řádku se přesunout do složky, ve které má být vytvořena aplikace:
 - `yo @microsoft/sharepoint`
 - Zvolit název aplikace, typ komponenty, název webových částí a framework aplikace React
 - Pro přidání dalších záložek (webparts) v příkazovém řádku ve vytvořeném SPFx solution spustit: `yo @microsoft/sharepoint` a přidat „webpart“
- Nainstalovat FluentUI pro React:
 - `npm install @fluentui/react` a `npm i @fluentui/react-northstar`

3. Vytvoření manifestu (kapitola 6.4):

- V Teamsech v Developer Portal vytvořit novou aplikaci:
 - V „Basic information“ zadat data totožná s vygenerovanou aplikací.
 - V „App features“ vytvořit „Personal app“ a zde vložit tolik položek, kolik je „webparts“.
 - V „App package“ zkontrolovat „Manifest“ soubor a stáhnout.
- Stáhnutý rozbalený soubor „manifest.json“ i s ikonkami vložit do složky „./teams“ v konkrétním SPFx projektu.
- Upravit kód programu:
 - V „manifest.json“ nahradit „contentUrl“ všech „webparts“:

```
"contentUrl": "https://{teamSiteDomain}/_layouts/15/TeamsLogon.aspx?SPFX=true&dest=/_layouts/15/teamshostedapp.aspx%3Fteams%26personal%26componentId={{SPFX_COMPONENT_ID}}%26forceLocale={locale}",
```

- Kde `{{SPFX_COMPONENT_ID}}` včetně obou závorek nahradit ID z konkrétní „webpart“ např. z „HelloWorldWebPart.manifest.json“

- V „manifest.json“ přidat do "validDomains" tyto domény:

```
"validDomains": [  
  "{teamsitedomain}",  
  "products.office.com",  
  "*.login.microsoftonline.com",  
  "*.sharepoint.com",  
  "*.sharepoint-df.com",
```

```
"spoppe-a.akamaihd.net",  
"spoprod-a.akamaihd.net",  
"resourceseng.blob.core.windows.net",  
"msft.spoppe.com"
```

]

- V „manifest.json“ přidat následující kód pro zajištění volání API v SPFx řešení:

```
"webApplicationInfo": {  
  "resource": "https://{teamSiteDomain}",  
  "id": "00000003-0000-0ff1-ce00-000000000000"  
}
```

4. Nahrání na SharePoint (kapitola 6.5):

- Ve složce „./teams“ vytvořit „zip“ soubor s názvem „TeamsSPFxApp“, jehož obsahem musí být „manifest.json“ a dva obrázky „outline.png“ a „color.png“.
- Ve vývojovém studiu spustit `gulp bundle --ship`, a pak `gulp package-solution --ship`
- Soubor „.sppkg“ ze složky „./sharepoint/solution“ nahrát na SP do „AppCatalog“:
 - Admin SP -> Aplikace -> Apps for SharePoint
 - Zobrazí se okno „Do you trust...“, které obsahuje „domains SharePoint Online“, odškrtnout „Make this solution available to all sites in the organization“ a „Deploy“
 - Synchronizovat s Teams: Files -> Sync to Teams
- Aplikace by se po synchronizaci měla objevit v Teams -> Další aplikace -> Vytvořeno pro vaši organizaci -> Přidat/Nainstalovat danou aplikaci po rozkliknutí

5. Testování (kapitola 6.6):

- Nainstalovat certifikát „localhost“ ve vývojovém prostředí VS Code v terminálu:
 - `gulp trust-dev-cert` a potvrdit instalaci certifikátu
 - Pokud by příkaz nefungoval, povolit spouštění skriptů: otevřít Windows PowerShell -> `Get-ExecutionPolicy -List` -> Pokud je u „CurrentUser“ nebo „LocalMachine“ nastaveno „AllSigned“ nebo „Undefined“, tak nastavit „RemoteSigned“: `Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine` (případně „CurrentUser“ místo „LocalMachine“)

- V config/serve.json nahradit kód:

```
"initialPage": "https://enter-your-SharePoint-  
site/_layouts/workbench.aspx"
```

kódem

```
"initialPage": "https://localhost:5432/workbench",  
"api": {  
  "port": 5432,  
  "entryPath": "node_modules/@microsoft/sp-webpart-  
workbench/lib/api/"  
}
```

- Testování přes SP Workbench:
 - V terminálu VS Code spustit `npm i @microsoft/sp-webpart-workbench` pro doinstalaci modulů
 - Spustit server `gulp serve`
 - Otevře se prázdná stránka localhost, ale místo ní otevřít odkaz, kde je název organizace, SharePoint web a doplnit adresu „_layouts/15/workbench.aspx“: „https://{tenant_name}.sharepoint.com/sites/{sharepoint_web_name}/_layouts/15/workbench.aspx“
 - Díky tomu bude možné pracovat s kontextem SharePoint, např. seznamy, knihovnamí atd.
 - Rozbalit SP pracovní plochu a vybrat dané řešení
 - Při každé úpravě kódu a následném uložení se změny ihned promítnou na otevřené SP webové stránce, což urychluje ladění
- Testování přes Teams
 - Spustit `gulp bundle` a `gulp package-solution` (bez `--ship`)
 - V příkazovém řádku by se mělo zobrazit "Warning - [package-solution] This is not a production build (--production), therefore the "includeClientSideAssets" setting will be ignored."
 - Balíček „.sppkg“ ze složky „./sharepoint/solution“ nahrát na SP do „AppCatalog“
 - Zobrazí se okno „Do you trust...“, které musí obsahovat lokální adresu „https://localhost:4321/dist“, odškrtnout „Make this solution available to all sites in the organization“ a „Deploy“

- Synchronizovat s Teams: Files -> Sync to Teams
 - Přidat a nainstalovat si aplikaci v Teams
 - V terminálu VS Code spustit `gulp serve`
 - Automaticky se otevře localhost stránka, ale místo ní otevřít přímo samotnou aplikaci v Teams
 - Po každé změně, která bude provedena ve VS Code během toho, co bude server spuštěn, se po uložení změny okamžitě nahrají a po znovunačtení stránky aplikace v Teams ihned zobrazí
 - Pro případné opětovné nahrání aplikace na SP bez testování, stačí u příkazů `gulp bundle` a `gulp package-solution` použít `--ship`
- Pokud se aplikace jednou nahraje na do „App Catalog“ na SharePointu a propojí se s Teamsem, je při znovu nahrání aplikace nejprve potřeba:
 - Odinstalovat přidanou aplikaci z Teams (není nutné)
 - Otevřít Teams jako správce -> Aplikace Teams -> Spravovat aplikace -> Najít požadovanou aplikaci -> Tři tečky -> Akce -> Odstranit
 - Otevřít „App Catalog“ jako správce na SharePointu a daný balíček „.sppkg“ odstranit
 - Poté je možné opětovně nahrát aplikaci

6. Konfigurace a další úpravy v aplikaci (kapitola 6.7):

- Ve všech „WebPart.manifest.json“ přidat následující kód nebo zkontrolovat, zda tento kód obsahuje, aby bylo umožněno automatické přidání aplikace do Teams po nahrání na SP „AppCatalog“, v opačném případě by se „zip“ soubor „TeamsSPFxApp“ musel ještě dodatečně nahrát na Teams, aby se spároval s řešením vloženým do „AppCatalog“ na SharePointu:

```
"supportedHosts": ["SharePointWebPart", "TeamsPersonalApp",
"TeamsTab", "SharePointFullPage"],
```

- V „config/package-solution.json“ přidat následující kód, pokud již neobsahuje. Hodnota musí být „true“, jinak nebude řešení dostupné v rámci celé organizace na Teams:

```
"skipFeatureDeployment": true,
```

7. Práce s daty (kapitola 6.8):

- SharePoint data (kapitola 6.8.1):
 - Instalace PnP verze 2 pro ukládání dat do seznamu (na verzi 3 nefunguje získávání položek ze seznamu na SP): `npm i @pnp/sp@2.11.0`
 - Import balíčku a získání položek SP seznamu:

```
import { Web } from "@pnp/sp";

const web = Web(this.props.webURL);
let items: any[] = [];
items = await web.lists.getByTitle("Attendance")
.items.select('*').get();
```

- Microsoft Graph data (kapitola 6.8.2):
 - Schválení oprávnění:
 - Přejít do sekce „Správa rozhraní API“ v Centru správy náhledu SharePoint: Správa -> SharePoint -> Upřesnit -> Přístup k API
 - Zde by neměla prozatím být žádná oprávnění.
 - Vytvoření oprávnění v Azure:
 - Přejít na stránku „Microsoft Entra ID“ na portálu Azure
 - V levém panelu kliknout na „Registrace aplikací“
 - Kliknout na „Zobrazit všechny aplikace“
 - Kliknout na položku „SharePoint Online Client Extensibility Web Application Principal“
 - V levém panelu zvolit „Oprávnění rozhraní API“
 - Kliknout na tlačítko „Přidat oprávnění“
 - V části „Vyberte rozhraní API“ zvolit „Rozhraní API Microsoft“
 - Zde vybrat „Microsoft Graph“ a „Delegovaná oprávnění“
 - V části „User“ vybrat potřebné oprávnění, např. „User.Read“ (Přihlášení a čtení profilu uživatele), kliknout na „Přidat oprávnění“
 - Kliknout na „Udělit souhlas správce pro ...“
 - Vrátit se zpět do sekce „Přístup API“ ve správě SharePointu
 - Zkontrolovat, zda je ve schválených oprávněních již uvedeno „Microsoft Graph“

- Vytvoření oprávnění v SPFx:
 - Do „package-solution.json“ vložit požadované oprávnění, např.:

```
"webApiPermissionRequests": [
  {
    "resource": "Microsoft Graph",
    "scope": "User.Read"
  }
],
```

- Připojení ke Grap API z SPFx:
 - Do interface „.ts“ souboru v „component“ vložit:

```
import { MSGraphClientV3 } from '@microsoft/sp-http';
client: MSGraphClientV3;
```

- V souboru „.ts“ do render vytvořit instanci:

```
client: await this.context.msGraphClientFactory.getClient('3')
```

- Komunikovat z „.tsx“ souboru:

```
const userInfo: any = await this.props.client.api('/users').version
('v1.0').get();
```

- Data Azure databáze (kapitola 6.8.3):
 - Vytvoření SQL databáze a SQL serveru v Azure Portal:
 - Na portálu Azure zvolit službu „SQL Database“ a nechat vytvořit novou databázi
 - V části „Základy“ vybrat předplatné a skupinu prostředků, vyplnit název databáze a zvolit existující server nebo vytvořit nový
 - Pro vytvoření nového serveru zvolit název serveru, umístění a metodu ověření (používat jak ověřování SQL, tak přes Microsoft Entra) a nastavit správce a přihlašovací údaje

- Nastavit úroveň výpočtů a uložení služby SQL podle potřeb a vybrat typ redundance uložení záloh
- V části „Sítě“ povolit veřejný koncový bod, umožnit službám a prostředkům Azure přistupovat k serveru SQL a přidat aktuální IP adresu klienta (lze upravit později)
- V části „Zabezpečení“ povolit identitu serveru (lze povolit později)
- Vytvoření webového rozhraní API ASP.NET Core:
 - V „Microsoft Visual Studio“ vytvořit nový projekt ze šablony „Webové rozhraní API ASP.NET Core“
 - V dalším kroku zvolit „.NET 7.0 Framework rozhraní“, nastavit konfiguraci na HTTPS, přidat předpřipravené rozhraní API a případně povolit podporu „OpenAPI (Swagger)“
 - Připravit si „Program.cs“ a „Startup.cs“ a rozdělit aplikaci na architekturu „Controllers“ a „Models“.
- Komunikace API s databází pomocí SqlConnection:
 - V okně „Průzkumník řešení“ kliknout na uzel „Závislosti projektu“ a vybrat „Spravovat balíčky NuGet“
 - Vyhledat a nainstalovat Microsoft.Data.SqlClient (pokud nebyla knihovna nalezena – v možnostech nastavení v sekci „Správce balíčků NuGet“ přidat do zdroje balíčků další balíček s názvem „nuget.org“ a se zdrojem „https://api.nuget.org/v3/index.json“
 - Z přehledu SQL databáze v sekci „Nastavení“ zkopírovat připojovací řetězec a vložit do .NET aplikace do souboru „appsettings.json“ jako „ConnectionString“:

```

"ConnectionStrings": {
  "AZURE_SQL_CONNECTIONSTRING": "Server=tcp:attendancesystem.
  .database.windows.net,1433;Initial Catalog=GuideTest;Encrypt=True;
  TrustServerCertificate=False;Connection Timeout=30;Authentication=
  \"Active Directory Default\";\"
}

```

- Vytvořit objekt v části „Models“:

```
public class MyConnectionModel
{
    public string AZURE_SQL_CONNECTIONSTRING { get; set; }
}
```

- Zavolat objekt v části „Controllers“:

```
public User(IOption<MyConnectionModel> azureString) {
    _azureString = azureString;
}

[HttpGet("GetUsers")]
public IEnumerable<string> Get() {
    string connectionString =
_azureString.Value.AZURE_SQL_CONNECTIONSTRING;
    using var conn = new SqlConnection(connectionString);
    conn.Open();
    ...
}
```

- Publikování webového rozhraní API ASP.NET Core do Azure App Service:
 - Do Microsoft Visual Studia se přihlásit účtem tenantu, přes který bude aplikace hostována
 - Kliknout pravým tlačítkem na projekt a dát „Publikovat“
 - V dialogovém okně zvolit „Azure“ a následně službu „Azure App Service (Windows)“ jako cíl nasazení
 - Vyplnit název aplikace, zvolit předplatné, skupinu prostředku a plán hostování (případně vytvořit nový), přeskočit krok API Management a kliknout na tlačítko „Dokončit“
 - Následně aplikaci nahrát přes tlačítko „Publikovat“
- Konfigurace zabezpečeného připojení mezi službou App Service a databází SQL pomocí spravované identity:
 - Na portálu Azure ve službě „App Service“ v levém navigačním panelu kliknout na „Identita“ a na kartě „Přiřazeno systémem“ nastavit stav na „Zapnuto“
 - Ve vytvořené databázi spravovat dotaz v části „Editor dotazů (Preview)“, přihlásit se přes účet správce nebo Microsoft Entra ID
 - Do editoru vložit následující skript s názvem vytvořené identity z „App Service“ pro vytvoření uživatele této služby v databázi:


```
CREATE USER attendancesystemapp FROM EXTERNAL PROVIDER;  
ALTER ROLE db_datareader ADD MEMBER attendancesystemapp;  
ALTER ROLE db_datawriter ADD MEMBER attendancesystemapp;  
ALTER ROLE db_ddladmin ADD MEMBER attendancesystemapp;  
GO
```

- Registrování aplikace v Microsoft Entra ID:
 - V „Microsoft Entra ID“ v levém navigačním panelu zvolit „Registrace aplikací“ a následně „Nová registrace“, vložit název nové aplikace
 - V registrované aplikaci v části „Zpřístupnit rozhraní API“ vytvořit identifikátor URI pro ID aplikace a přidat obor „access_as_user“
 - V registrované aplikaci jako službě „SharePoint Online Client Extensibility“ v části „Oprávnění rozhraní API“ kliknout na „Přidat nové oprávnění“ v části „Rozhraní API, která používá moje organizace“ se odkázat na vytvořený obor definovaný rozhraním API nově registrované aplikace a udělit souhlas správce
- Nastavení CORS pro přístup ke zdrojům v jiné doméně:
 - Ve službě „App Service“ v části „CORS“ kliknout na „Povolit“ a přidat URL adresu, na které bude hostovaná SPFx aplikace
- Komunikace SPFx s webovým rozhraním pomocí AadHttpClient:
 - Do SPFx řešení importovat „AadHttpClient“, vytvořit novou instanci a předat zdroj pro připojení, což je ID klienta nově registrované aplikace v „Microsoft Entra ID“
 - ID klienta je předáno jako druhý parametr v konstruktoru „AadHttpClient“ jako GUID
 - Zadat URL a webový požadavek pro komunikaci s API v „Azure App Service“

```

import { AadHttpClient, HttpClientResponse } from '@microsoft/sp-
http';

private async loadUsers():Promise<void>{
    this.props.webpartcontext.aadHttpClientFactory.getClient('api://
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx')
    .then((client: AadHttpClient): void => {
        client.get('https://attendancesystemapp.azurewebsites.net/api/
User/Users', AadHttpClient.configurations.v1)
    .then((response: HttpClientResponse): Promise<any[]> => {
        return response.json();
    })
    .then((users: string[]): void => {
        console.log(users)
    })
    .catch((e: Error) => console.log(e));
})
.catch((e: Error) => console.log(e));
}

```

- Nastavení zabezpečeného přístupu k API pomocí API Scope:

- U webového rozhraní „API ASP.NET“ přidat konfiguraci pro autentizaci a autorizaci
- Do souboru „appsettings.json“ přidat vlastnost „Azure AD“, která obsahuje instanci, doménu, tenantId, clientId, cestu pro přesměrování a rozsah oprávnění a následně znovu publikovat:

```

"AzureAd": {
    "Instance": "https://login.microsoftonline.com/",
    "Domain": "xxxxxxxx.onmicrosoft.com",
    "TenantId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx",
    "ClientId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx",
    "CallbackPath": "/signin-oidc",
    "Scopes": "access_as_user"
},

```