

ŠKODA AUTO VYSOKÁ ŠKOLA o.p.s.

Studijní program: Podniková ekonomika a manažerská informatika

**Návrh a vývoj aplikace v prostředí
Microsoft Power Platform
Bakalářská práce**

Uladzimir Kalougin

Vedoucí práce: Ing. Lukáš Herout, Ph.D



ŠKODA AUTO Vysoká škola

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Zpracovatel: **Uladzimir Kalougin**

Studijní program: Podniková ekonomika a manažerská informatika

Název tématu: **Návrh a vývoj aplikace v prostředí Microsoft Power Platform**

Cíl: Návrh a realizace aplikace pro zvýšení efektivity pracovních procesů v podniku ŠKODA AUTO a.s. a to s využitím nástrojů Microsoft Power Platform (Power Apps, Power Automate a Power BI).

Rámcový obsah:

1. Úvod do problematiky návrhu a vývoje aplikací.
2. Seznámení s nástroji Power Platform a srovnání s jinými programovacími nástroji/prostředími.
3. Analýza a výběr procesu, popis životního cyklu.
4. Návrh a realizace aplikace. Vyhodnocení přínosů.
5. Možnosti dalšího efektivního využití Power Platform v rámci ŠKODA AUTO a.s.

Rozsah práce: 25 – 30 stran

Seznam odborné literatury:

1. PETROV, A. *Database internals: a deep dive into how distributed data systems work*. Beijing: O'Reilly, 2019. ISBN 978-1-4920-4034-7.
2. SUDARSHAN, S. – SILBERSCHATZ, A. – KORTH, H. *Database System Concepts*. New York: Osborne-McGraw-Hill, 1997. ISBN 0-07-044756-X.
3. Microsoft a.s.: Microsoft Docs: Power Apps, Power BI, Power Automate

Datum zadání bakalářské práce: prosinec 2021

Termín odevzdání bakalářské práce: prosinec 2022

L. S.

Elektronicky schváleno dne 26. 5. 2022

Uladzimir Kalougin

Autor práce

Elektronicky schváleno dne 26. 5. 2022

Ing. Lukáš Herout, Ph.D.

Vedoucí práce

Elektronicky schváleno dne 27. 5. 2022

prof. Ing. Jiří Strouhal, Ph.D.

Garant studijního programu

Elektronicky schváleno dne 27. 5. 2022

doc. Ing. Pavel Mertlík, CSc.

Rektor ŠAVŠ

Děkuji Ing. Lukáši Heroutovi Ph. D. za odborné vedení závěrečné práce, za jeho trpělivost a zpětnou vazbu.

Obsah

Úvod.....	8
1 Teoretická východiska řešení	9
1.1 Zásady návrhů aplikací	9
1.1.1 Rozdíl mezi procesem a projektem.....	9
1.1.2 Životní cyklus projektu	11
1.2 Zásady realizaci aplikací	11
1.2.1 Princip malých aplikací	11
1.2.2 Princip malých datových sad	12
1.2.3 Princip omezení času	12
1.3 Analýza vývojového prostředí Microsoft Power Platform.....	14
1.3.1 Microsoft Power Automate.....	14
1.3.2 Microsoft Power Apps.....	15
1.3.3 Microsoft Power BI.....	15
1.3.4 Microsoft Power Virtual Agents.....	15
1.3.5 Microsoft Power Pages	16
1.4 Databáze a datové sady.....	16
1.4.1 Entity-Relationship model	16
1.4.2 Azure SQL Database	19
1.4.3 SharePoint.....	19
1.4.4 Excel.....	19
2 Analýza současného stavu	20
2.1 Analýza existující aplikace	20
2.2 Analýza vybraného řešení	20
2.3 Základní funkce prostředí Microsoft Power Platform	21
2.3.1 Spojení sady dat s galerií.....	24
3 Vlastní aplikace.....	26
3.1 Návrh aplikace	26
3.2 Realizace aplikace I	27
3.2.1 Automatizace periodické změny dat	27
3.2.2 Programování a algoritmizace požadovaných funkcí.....	28
3.2.3 Realizace UI	33
3.3 Realizace vizualizací pomocí Power BI.....	34
3.4 Výsledky realizace.....	34

4	Možnosti dalšího využití Power Platform v rámci ŠKODA AUTO a.s.....	36
	Závěr	37
	Seznam literatury	38
	Seznam obrázků a tabulek.....	41
	Seznam příloh	43

Seznam použitých zkratk a symbolů

DBMS Database Management System

IS Information System

IT Information Technology

ICT Information and Communication Technologies

JSON JavaScript Object Notation

SaaS Software as a Service

PaaS Platform as a Service

UI User Interface

Úvod

Velký počet společností používá různé aplikace pro různé cíle, jako evidence jednotek, požadání a schválení projektů, logování veškerých dat atd. Obvykle takové programy mohou být podporované IT oddělením společnosti, nebo externím způsobem tak, že aplikace mohou být napsány ve různých programovacích jazycích. To většinou vede k tomu, že pro podporu aplikace společnosti obrátí ke specialistům zabývajícím programovacími prostředí, na kterých byli vyvinuté ty programy, co vede k zvětšení počtu lidí. Z toho důvodu takové programy většinou potřebují velkou finanční podporu zvyšující celkové náklady.

Microsoft Power Platform povoluje pracovníkům sledovat, podporovat a vyvíjet svoje osobní aplikace a toky, které mohou používat pro automatizace a zjednodušení jejich práci. Automatizace může vest ke zrychlení práce, protože zaměstnanec bude moci provádět jiné části své práce ve stejný čas, kdy se budou například připravovat data podle automatizovaných toků, takže se sníží náklady společnosti, protože zaměstnanci mohou sami opravovat a podporovat aplikace.

Microsoft Power Platform se skládá z pěti služeb, Power Apps, Power Automate, Power BI, Power Pages a Power Virtual Agents. Každý je nástrojem, který může být použit samostatně, ale při spojení bude využit jejich maximální potenciál. Platforma pomáhá uživatelům při vyvíjení aplikací, automatizovaných procesů, umělých chatbotů, vizualizací dat a při tvoření low-kóde webových stránek.

V této bakalářské práci, bude popsán nástroj pro automatizace a konstruování aplikace prostředí Microsoft Power Platform, a jeho srovnání s jinými vývojovými prostředími v oblasti informačních technologií. Dalším bodem bude vyvinutý program a automatizované toky, které zjednoduší vybrané části byrokratické práce ve společnosti ŠKODA AUTO a.s.

Práce se soustředí na vysvětlení pojmu týkající se oblastí IT a projektového managementu a jejich porovnání s existujícími metodami na pracovišti. Téma této bakalářské práce bylo vybráno kvůli vysokému potenciálu vývojového prostředí Microsoft Power Platform.

1 Teoretická východiska řešení

Tato kapitola bude zahrnovat všechna teoretická východiska, která budou sloužit k lepšímu porozumění problematice bakalářské práce. Všechny základní definice a principy, které jsou dále uvedeny, byly spojené s vývojem aplikace a měly vliv na dosažení stanovených cílů projektu.

1.1 Zásady návrhů aplikací

Návrh je neoddělnou součástí libovolného projektu. V této fázi řeší, co se od projektu vyžaduje, jaké budou výhody a nevýhody, co je potřebou na vstupu a jaký bude výstup. Do návrhu aplikací se může přidat procesní analýza, která je jednou z nejdůležitějších analytických technik, popisující tok práce, zlepšující výkonnost, účelnost, efektivnost, hospodárnost nebo profitabilitu (ManagementMania, 2018).

Podle procesní analýzy je možné popsat vývoj aplikace takto:

- Časové ohraničení vývoje,
- Obsah a cíle projektu,
- Určení vstupu a výstupu projektu,
- Zjistit rizika projektu,
- Vytvářet kritickou cestu a následně ji optimalizovat.

Po etapě konsolidací procesní analýzy, bude začínat praktická část vývoje, která čtenáře seznámí s vývojovým prostředím, jeho programovacím jazykem, výběrem datové sady, a nakonec i tvořením UI a zdrojového kódu.

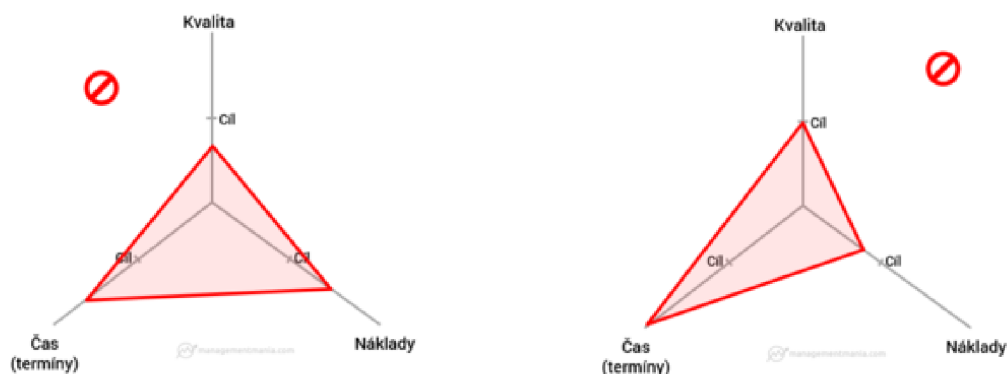
1.1.1 Rozdíl mezi procesem a projektem

Proces je opakovatelný děj, který se skládá z uspořádaných akcí, má začátek i konec, a při vynaložení materiálních, lidských, finančních a informačních vstupů, pomocí řízení může vést k požadovanému a očekávanému výstupu, mající hodnotu pro zákazníka (Šmíd, 2007).

Podle Doležala, Máchala a Lacko (2012) je projekt proces, který není opakovatelný. Je časově, zdrojově a finančně omezený, a jeho implementace směřuje k požadovaným výstupům v nutné kvalitě a při splnění požadovaných standardů.

Obvykle společnost chce, aby byl projekt udělán co nejrychleji, nejlevnější cestou a s nejlepší kvalitou. Ale to je nereálné. Projekt může být vytvořen za nízké náklady

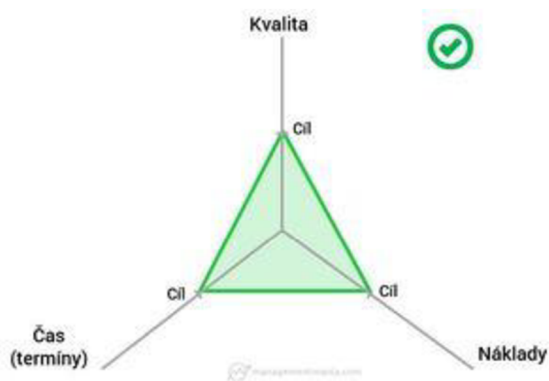
a za krátký časový úsek, avšak kvalita takového projektu bude nízká. To se může stát s každou proměnnou projektového trojúhelníku, který je závislý jeden na druhém (viz Obr. 1).



Zdroj: Magický trojúhelník projektového řízení, 2019

Obr. 1 Špatně řízené projekty

Proto projektový manažer musí najít rovnováhu mezi kvalitou, časem a náklady, pro dosažení co nejlepšího výsledku (viz Obr. 2).



Zdroj: Magický trojúhelník projektového řízení, 2019

Obr. 2 Dobře řízený projekt

1.1.2 Životní cyklus projektu

Jak bylo napsáno výše, projekt má svůj omezený čas na splnění, který tvoří životní cyklus. Podle Univerzity Projektového Managementu (2017) životní cyklus projektu se obvykle skládá ze čtyř fází:

- I. Začátek/iniciace projektu,
- II. Organizace a příprava řešení,
- III. Realizace,
- IV. Konec projektu a zavedení do užívání.

Začátek, anebo iniciace projektu, je obvykle podporován Zakládací listinou, která obsahuje hlavní charakteristiky projektu, jako název, zadavatele, cíle, role a jejich zodpovědnosti, hrubý finanční plán, časový plán a předpokládanou vazbu na jiné projekty. Organizace a příprava obsahuje strategie a návrhy řešení na splnění plánu, návrh designu, koncepty, označení teamu a jejich zodpovědnosti a pokud je potřeba, tak i zabezpečení vnějších dodávek. Fáze realizace zajišťuje veškeré praktické rozhodnutí a development, pomocí kterých se budují teamy na základě předchozích fází. Konec projektu je posledním stadiem, které směřuje k zavedení hotového produktu do provozu. Stádium organizace a realizace jsou navázány na sebe a mohou být cyklické, při takových situacích, kdy dojde k odkrytí nových vlastností nebo problémů při realizaci, nebo objevení nových nápadů.

1.2 Zásady realizaci aplikací

Vývoj aplikací, které jsou potřebné v současné době pro podniky, je komplikovaný děj. Přístup k němu se mění každý rok. Většina aplikací byla velkou hromadou funkcí, napojené v několika tisících řádků kódu. V dnešní době je to může být jinak. Určitě existují velké aplikace, které zahrnují širokou funkcionalitu a mají velký počet developerů pracujících nad nimi, ale v této bakalářské práci se zabýváme aplikacemi, které může vyvinout i jeden člověk, a které se mohou stát velkou podporou pro kancelář i zaměstnance.

1.2.1 Princip malých aplikací

První princip vývoje je princip malých aplikací, pomocí něhož se nepíše velký program, ale obsahuje všechny požadované funkce pro splnění různých úkolů:

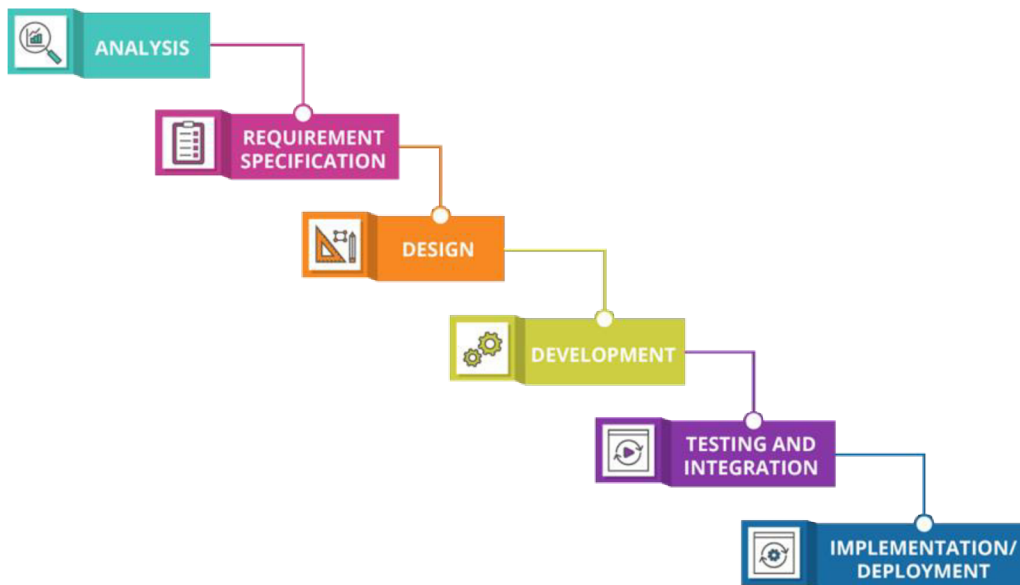
možnost napsat malé programy a přidělit jim potřebné funkce pro každou pracovní pozici, přičemž všechny ty malé aplikace mohou být napojené navzájem v datových sadách. Tím velice sníží tlak na team vývojářů, což bude mít pozitivní vliv na projektu. Stetson píše (2018), že snížení kognitivní zátěže vývojářů je prioritou, protože se pak mohou soustředit na řešení problému, místo aby měli na paměti aktuální komplexní model celé aplikace a vyvíjené funkce.

1.2.2 Princip malých datových sad

Další princip se skládá z toho, že Power Platform není příliš vhodný pro práci s velkými datovými sadami. Proto se velký počet dat bude zpracovávat mnohem pomaleji než malý počet dat. Tudíž kdyby byla potřeba uložit data, která už byla nepotřebná v aplikaci, musí být zavedeno automatizované logování dat do souborů na cloudové službě, které použité v aplikaci. Tím bude docíleno vyššího výkonu aplikace.

1.2.3 Princip omezení času

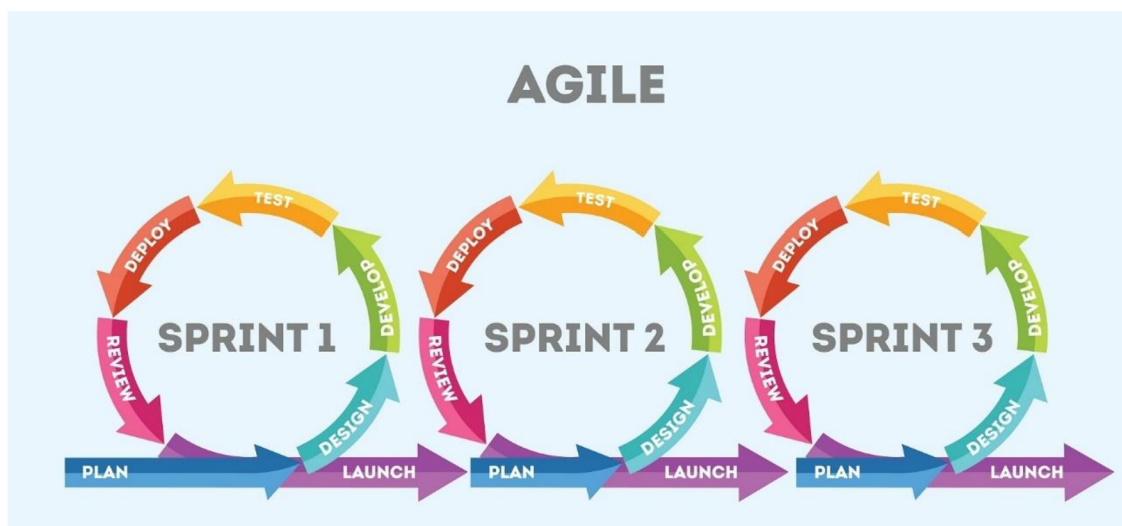
Třetí zásada je omezení času vývoje aplikace. Podle Stetsona (2018) už je to dávno, kdy vodopádový přístup řízení projektů (anglicky Waterfall) (viz Obr. 3) byl pouze jedním standardem vývoje aplikace. Tehdy byla vývojová doba velké aplikace od půl roku do dvou let. Vývojáři museli číst veškeré informace ohledně projektu: architekturu celé aplikace, dokumentace stávajícího systému, požadavky k produktu – a pak vytvarovat model, pomocí kterého psali kód. Časem se požadavky a s tím i architektura měnila, ale týmy vývojářů nedostávaly potřebné informace včas. I při menším požadavku museli projít všemi stádii přístupu, aby provedli nutné změny. Při tomto přístupu byla nejhorším scénářem paralyzace celého projektu.



Zdroj: Jones a Waddell, 2019

Obr. 3 Metoda Waterfall

V současné době se stalo největší změnou zaměřování se ne na „těžký“ program, ale na malé části kódu konkrétních funkcí, které mohou být hotové za dobu dvou až třech týdnů. Největší podporou změny se stal agilní přístup řízení projektu (viz Obr. 4), což vedlo ke snížení tlaku na vývojáře a zaměření se na postupné obnovení programu.



Zdroj: Slawek-Polczynska, 2020

Obr. 4 Metoda Agile

1.3 Analýza vývojového prostředí Microsoft Power Platform

Microsoft Power Platform je jednoduchým nástrojem pro všechny, kteří neznají programování, protože je možné částečně tvořit jednoduché algoritmy v Power Automate i bez použití kódu, nebo takzvaného low-kódu v Power Apps. Platforma může být ale i komplexním nástrojem, pro lidi, kteří znají programování, protože se může spojit systém s SQL servery a dalšími databázemi a využívat různé pokročilé algoritmy.

Celkem Microsoft Power Platform zahrnuje velkou sadu funkcí pro automatizaci byznys procesu, ale i pro tvorbu aplikací a vizualizací výsledku vhodnou pro analýzu. Marek (2020) uvádí, že všechny nástroje se spojují mezi sebou v komplexní systém, ve kterém jsou různé procesy vzájemně propojené. Jednou z hlavních výhod je to, že platforma umožňuje velmi rychlé a jednoduché sdílení dat mezi aplikacemi, bez použití dlouhé a složité integrace dat, kterou by bylo nutné použít při užití jiných specializovaných ICT.

1.3.1 Microsoft Power Automate

Automatizace je procesem, který začíná být v současné době nezbytným prakticky na každém pracovišti. Je potřebná k uvolnění části času, který je odváděn na rutinní děje, jako pracování s formuláři, stejnými výpočty, pracování s velkými datovými sadami atd., takže automatizace snižuje tlak na zaměstnance, a on se díky tomu může zabývat více prioritní prací. Z toho důvodu společnosti nebudou potřebovat zaměstnávat nové lidi pro tyto činnosti.

Power Automate je nástrojem nejenom pro automatizace různých byznys procesů a obchodní logiky, ale i pro tvorbu aplikačních funkcí. Jde o komplexní způsob, který umožňuje spojit více než sto různých druhů konektorů/servisů pro dosažení nutných výsledků. Při správném použití a napojení na Power Apps to přináší velmi širokou sadu různých funkcionalit, které mohou být spojené s různými cloudovými schránkami a datovými servery.

Automatizace se může týkat nejenom složitých procesů jako uložení souboru na cloudových úložištích pomocí formátování a integrace dat, ale i každodenních procesů, jako upozornění při změně souboru, automatizované připomínky, informování teamu, uložení příloh z mailu, logování procesů atd.

1.3.2 Microsoft Power Apps

Podle Bergera (2020) Power Apps je webovým programem pro rychlé a snadné vytvoření různých podnikových aplikací. Cílem tohoto informačního systému je tvoření webových nástrojů pro digitalizaci, sjednocení a automatizaci vnitropodnikových procesů pro zrychlení a zjednodušení prací zaměstnanců.

Power Apps může spojit s velkou řadou datových zdrojů, jako Azure SQL database, SQL servery, SharePoint, Excel a další. S tímto nástrojem můžeme použít automatizované toky, vizualizace dat v reálném čase, digitální podepisování, vložení souboru a práce s nimi. Nástroj povoluje, nejenom vyvíjet jednoduché aplikace pomocí low-kódu, ale i komplexní programy s použitím, toku, komponent a programování algoritmu. Jedním z omezení práce v Power Apps je to, že nástroj se nedoporučuje používat s velkými datovými sady o více jak 5000 položek, protože takovým způsobem rychlost výkonu aplikací bude mnohem nižší, a mohou se objevovat chyby.

1.3.3 Microsoft Power BI

Power BI je mocným nástrojem pro vizualizace a analýzu dat z různých datových zdrojů. Umožňuje vytvářet souborné reporty, které mohou velmi pomoci při rozhodnutí, takže nástroj umožňuje jednoduché integrování do známého prostředí Microsoft 365 (Berger a Hlavinka, 2020).

Power BI umožňuje jednoduché sestavení tabulek, webových stránek s vizualizacemi, snadnou práci s daty, grafické znázorňování, a ještě má velkou sadu různých nástrojů. Lze tedy propojit Power BI s daty, které se nachází v zdrojích různých databází.

1.3.4 Microsoft Power Virtual Agents

Power Virtual Agents povolí uživatelům vytvářet pro různé potřeby chatboty s umělou inteligencí, které mohou odpovídat jak na jednoduché požadavky, tak i na komplexní, požadující složitější komunikaci. Služba podporuje množství různých jazyků, a celou řadu různých webových stránek, mobilních aplikací a sociálních sítí které jsou podporovány Azure Bot Framework (Přehled Power Virtual Agents, 2022).

1.3.5 Microsoft Power Pages

Power Pages je SaaS (Software as a Service) platforma podnikové úrovně, která povoluje pomocí malého množství kódu vytvářet moderní externě orientované obchodní weby z možností jejich hostování a řízení. Platforma nabízí bohaté šablony s možnostmi přizpůsobení, plynulé vizuální prostředí, které vám umožní rychle vytvořit webové stránky odpovídající potřebám vaší firmy (Co je Power Pages, 2022).

1.4 Databáze a datové sady

Pro většinu existujících webových aplikací platí, že mají nějaký databázový systém, ve kterém jsou uloženy sady dat. Firma může používat data pro podporu aplikace, pro statistickou analýzu nebo jako součást systému. Microsoft Power Platform není výjimkou. Pro tvorbu aplikace, je nutno vytvořit databázový systém.

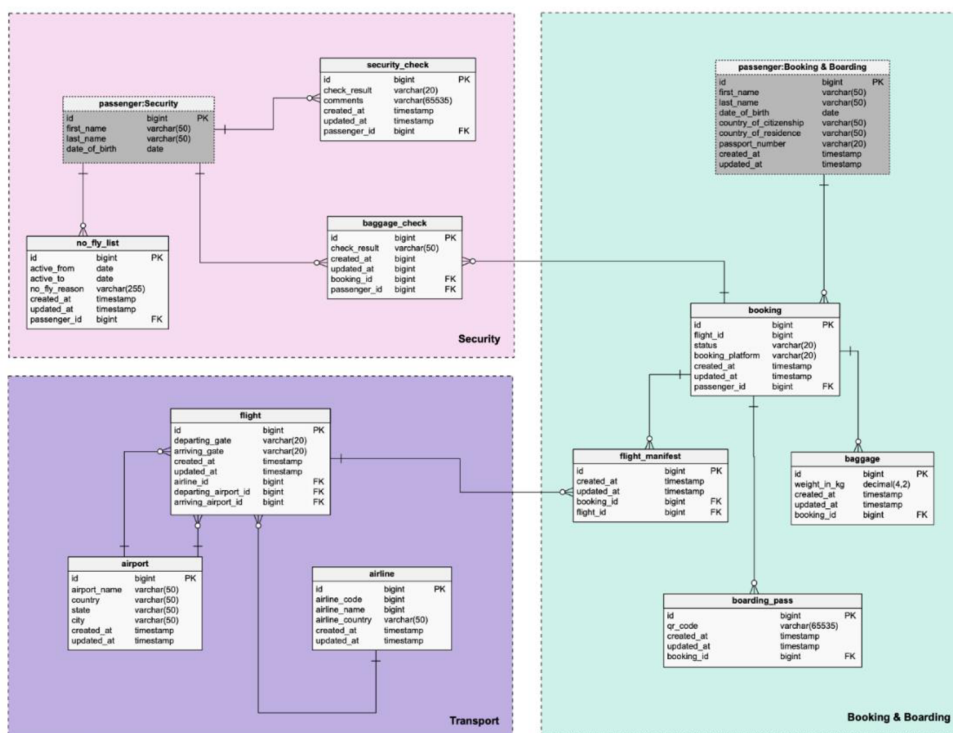
Dle Petrova (2019) hlavními funkcemi DBMS (database management system) jsou chránit data a umožňovat jejich poskytování uživatelům. Jako primární zdroj dat používané databáze, které pomáhají vyměňovat data mezi částmi aplikace. Aby vývojáři nehledali nebo nevyvíjeli nové způsoby ukládání, organizace a načítání informací při každém vývoji nové aplikace, jsou používané databáze. Tím způsobem vývojáři mohou soustředit na logiku aplikace než na její infrastrukturu.

Databáze jsou uspořádané sbírky strukturovaných dat, které se obvykle ukládají v elektronickém formátu v počítačovém systému. Databáze jsou obvykle kontrolovány databázovým řídicím systémem, který spolu s daty a aplikacemi pomáhají data řídit. Ve většině případů, jsou databáze strukturované pomocí jazyka SQL, který umožňuje modifikace, tvoření, řízení a kontrolu dat (What Is Database, 2020). Datová sada je uskupení pojmenovaných dat, které odkazují na data, nutná pro používání v aplikacích, jako vstupy a výstupy. Obvykle jsou strukturované pomocí tabulek, sloupců a řádků.

1.4.1 Entity-Relationship model

Entity-Relationship model nebo ERM byl vyvinut s cílem usnadnit návrh databáze pomocí náčrtu, které představuje celkovou logickou strukturu databáze. E-R model je velmi užitečný nástroj pro mapování libovolných dějů a jejich vizualizací pomocí konceptuálních schémat. Model obsahuje tři základní koncepty: sady entit, sady vztahu a atributy. Takže model obsahuje grafické znázornění vztahu entit, který se

nazývá E-R diagram (viz Obr. 5). Diagram je jednoduchý pro porozumění, díky čemuž je model široce používán (Silberschatz, Korth, Sudarshan, 2020, str. 244).



Zdroj: Rathee, 2021

Obr. 5 Příklad uspořádaného E-R diagramu

Sada entit je sadou subjektů stejného druhu obsahující stejné vlastnosti nebo atributy. Například všichni vývojoví manažeři ve společnosti, mohou být zapsaní do jedné entitní sady Development Manager. Testery na pracovišti mohou být přidány do entitní sady Tester.

Každá sada subjektů vlastní sady atributů, které popisují vlastnosti každého člena množiny entit. Když je atribut uveden pro množinu entit, znamená to, že databáze má podobnou informaci pro každý její člen, ale to však neznamená, že atribut obsahuje stejnou informaci pro každou entitu (Silberschatz, Korth, Sudarshan, 2020, str. 245). Hlavním atributem, který je přiřazen databázovým systémem bez účasti programátoru, je ID, podle něho můžeme najít jakoukoliv informaci v sadě entit, i kdyby by byla jiná informace neznámá (Obr.6).

Development Manager			
ID	Jméno	Přimění	Tel. Číslo
1	Honza	Košár	775 475 xxx
2	Jozef	Halík	773 186 xxx
3	Radim	Merund	775 990 xxx

Tester			
ID	Jméno	Přimění	Tel. Číslo
1	Vojta	Stálý	775 123 xxx
2	Adam	Malý	773 988 xxx
3	Vladislav	Kostel	773 444 xxx

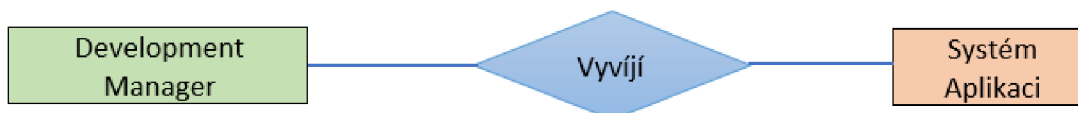
Obr. 6 Příklad sad entit

Navzájem se sadami entit, existují i sady vztahu. Vztahy jsou asociativní relace mezi různými druhy entit (Silberschatz, Korth, Sudarshan, 2020, str. 246). Příklad: pro vývojového manažera Halíka, který je odpovědný za vývoj Front endu, můžeme přiřadit vztah *vyvíjí*, který bude spojovat Front end a Halíka (viz Obr. 7). Tento vztah bude znamenat, že vývojový manažer s příjmením Halík vyvíjí Front end aplikaci. Sada vztahu je kolekce vztahu stejného typu (viz Obr. 8).

Development Manager			
ID	Jméno	Přimění	Tel. Číslo
1	Honza	Košár	775 475 xxx
2	Jozef	Halík	773 186 xxx
3	Radim	Merund	775 990 xxx

Systém Aplikaci	
ID	Součást systému
1	Back end
2	Front end
3	UX/UI

Obr. 7 Příklad relace mezi entitami



Obr. 8 Příklad sady vztahu

1.4.2 Azure SQL Database

Azure SQL database je jedním z nástrojů, které lze připojit k Microsoft Power Platform, jako databázový systém. Azure SQL je databázovým strojem typu PaaS (Platform as a Service), která zahrnuje funkce aktualizace, opravy, zálohování a monitorování bez nutnosti dějů ze strany uživatele, takže SQL Database může povolit zpracovávat relační data i nerelační, jako grafy, JSON, prostorové a XML (Co je Azure SQL Database, 2022). Z důvodů toho, že Azure SQL Database není součástí Microsoft 365, pro jeho používání společnost musí nakoupit licenci pro zaměstnance, ale plusem je to, že SQL databáze je mnohem výkonnější než ostatní služby, které mohou být spojeny s Power Platform.

1.4.3 SharePoint

Microsoft Power Platform umožňuje spojit své nástroje se SharePointem, a používat je jako databáze. SharePoint je jedním z nástrojů Microsoft 365, který umožňuje tvoření webových stránek pro společenskou teamovou práci. Weby mohou být použité, jako bezpečné datové schránky a úložiště souborů, ve kterých je možnost manipulovat daty (Začínáme s SharePointem, 2022). Cílem tohoto nástroje je zjednodušit teamovou práci a dostupnost dat pro členy teamu. SharePoint může být použit, jako databázový systém, ve kterém jsou uloženy datové seznamy a knihovny souborů, jako úložiště dokumentů pro logování dat. Minusem služby je to, že SharePoint nebyl vytvořen, jako databáze, ale plusem je to, že je možné ho použít tímto způsobem (Baizini, 2017). Dalším velkým plusem je jeho jednoduché použití společně s Power Apps, jakož i to, že je součástí Microsoft 365, a objednávat licence zvlášť není potřeba.

1.4.4 Excel

Excel lze použít spolu s Microsoft Power Platform, jako jeden z konektorů v Power Automate, nebo jako datovou sadu v Power Apps, takže soubory s formátem Excelu, je možné velmi jednoduše integrovat do SharePointu. Excel je tabulkový procesor, který pomáhá pracovat s tabulkami a daty, výpočty a analýzou dat, takže pomocí různých funkcí umožňuje snadné filtrování, řazení a upravování dat, jakož má i nástroje pro vizualizaci informací ve formě veškerých grafů (Microsoft Excel, 2022).

2 Analýza současného stavu

V této kapitole bude popsána analýza současného stavu aplikace pro založení, řízení a manipulaci se stávajícími projekty, kterými se zabývá oddělení EEI ve společnosti ŠKODA AUTO a.s., takže bude popsáno vybrané řešení pro novou aplikaci a základy používání Microsoft Power Apps.

2.1 Analýza existující aplikace

V současném stavu na pracovišti existuje byznys aplikace TIMON pro manipulace, zavedení a řízení řídicími jednotkami souvisejícími s jednotlivými automobily, která se staví v programovacím prostředí Excel VBA Editor. Aplikace byla vyvinuta na pracovišti teamem odpovědným za IT rozvoj oddělení.

Aplikace je komplexním systémem a zahrnuje v sobě velkou sadu funkcí, podporou aplikací a vyvinutí pro dalších funkcí, se zajímá team skládající se ze tří lidí. Vývoj prochází agilní metodou. Daty, které jsou uloženy na virtuálním disku společnosti je ve formátu .csv.

Aplikace TIMON umožňuje zavádět nové řídicí jednotky a funkce pro různé vozy, sledovat a ověřovat jejich stav. Obsahuje systém evidence řídicích jednotek, testy pro adaptaci jednotek v nových vozech a ověřování a otestování jejich funkcí. Ve funkcionalitě aplikací existuje systém testovacích termínů a termínů vydání.

Obrázky stávající aplikace budou v Příloze č.1.

2.2 Analýza vybraného řešení

Z důvodu toho, že aplikace je používána na pracovišti už pět let, a její systém a UI (User interface) se neměnil, byl zaveden úkol – navrhnout nový systém manipulace s daty, který bude používat moderní vývojové prostředí Microsoft Power Platform, jakož i provést analýzu, podle které se bude na konci projektu řešit užívání platformy v dalších oblastech na oddělení. I když stávající aplikace je snadný nástroj a při její užívání se nevyskytly žádné problémy, stále je mínusem to, že potřebuje více času na vyvinutí nových funkcí.

Prostředí Power Platform bylo vybrané, mezi ostatní automatizační ICT (Information and Communication Technologies), protože ve společnosti ŠKODA AUTO a.s. je velice používaná služba Office 365, a další služby od společnosti Microsoft. Proto by to umožnilo rychlejší a jednodušší integraci Platformy a jejích systémů do

používání, jakož i integraci už existujících dat do nových aplikací. Velkým plusem Power Platform se stalo, že nejenom umožní poměrně rychlé vyvinutí potřebných aplikací, ale i povoluje lepší integrace dat do veškerých součástí platformy. Znamená to, že lze manipulovat s daty, automatizovat jejich integraci do systému a vizualizovat je v reportech Power BI.

Jako databázový systém byl použit SharePoint, jeho webové stránky obsahují knihovny souborů a seznamy, které lze jednoduše připojit k Power Apps. I když je nedostatkem to, že SharePoint nebyl zkonstruován primárně jako databáze, ale je pro to vhodný. Další příčinou, proč byl použit SharePoint je to, že je součástí Microsoft 365 a proto nemusí oddělení žádat o doplňující licenci, takže je jednoduchý v používání spolu s celou Power Platformou.

V případě rychlé implementace nového IS do pracovního procesu, bude spojeno s chybami a snížením efektivity prací na počátečních etapách, kvůli odporu zaměstnanců vůči novému systému (Bednař, 2022, str. 2). Na konci integračního procesu nového systému a adaptaci zaměstnanců, však lze výkon zlepšit a velký obsah prací zjednodušit. Proto musí být postavení nového systému plynulé a integrace nesmí ovlivnit pracovní proces, protože to může vést k snížení zisku společnosti.

2.3 Základní funkce prostředí Microsoft Power Apps

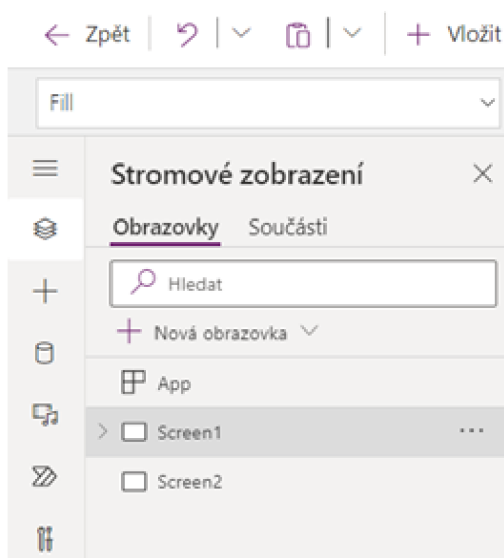
Na začátku prací s nástrojem Power Apps je možnost vybrat jednu z existujících šablon aplikací, které mají už nastavenou a popsanou funkcionalitu nebo začít z prázdného plátna, ve kterém všechno budeme nastavovat sami. Obě varianty jsou velice používané a ani jedna z nich nebude zdržovat kreativitu vývojáře, protože každý prvek, který se bude dodávat nebo už existovat, je možné změnit. Ale v této práci bude používané prázdné plátno. V Power Apps je možnost výběru formátu aplikace, která umožní vývoj nejenom počítačových ale i mobilních aplikací.

První věc, kterou uvidí vývojář, je pracovní prostředí nástroje Power Apps. Na levé straně lze vidět stromové zobrazení, kde jsou ukázané dvě varianty:

Obrazovky – tady budou zobrazené veškeré prvky budoucí aplikace a její obrazovky, mezi kterými, je možnost se pohybovat pomocí stromového zobrazení, nebo pomocí prvku s algoritmem, který bude ukazovat na jinou obrazovku,

Součásti – při tomto zobrazení Power Apps přepne na jiné zobrazení, kde existuje nástroj, který povoluje vyvíjet své komponenty s funkcionalitou, kterou bude

nastavovat vývojář a pak může používat v aplikaci. V případě těchto prací nástroj *Součásti*, bude používán jenom při realizaci UI, protože při změně komponenty se budou měnit všechny prvky na každé obrazovce, kde byli použity. Je to velmi zajímavý a užitečný prostředek, který může pomoci ve velkém počtu situací, kde bude potřeba udělat specifickou komponentu přímo pro ukázanou funkci. Tyto prvky nejsou lepší variantou při procesu vývoje, kde je potřeba rychle reagovat na požadavky, jakož i neobsahují velkou sadu funkcí, která je u už existujících prvků, což je velkým omezením.



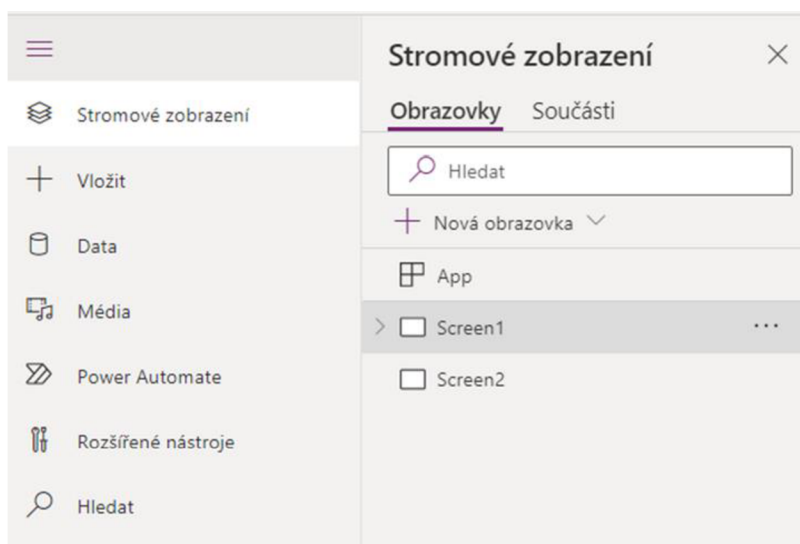
Obr. 9 *Obrazovky a součásti*

Vlevo je okno s nabídkou dalších funkcí (viz Obr. 10), které můžeme vybrat a používat:

- *Vložit* – tady se nachází veškeré prvky, které můžeme použít při vývoji aplikace,
Data – položka, která umožňuje podívat se na datové sady, které budou navazovat na prvky aplikace,
- *Media* – obrázky, video, zvuk, všechny multimediální soubory, které budou součástí aplikace se nachází tady,
- *Power Automate* – tady budou ukázány toky které pomohou v automatizaci nebo realizaci některých funkcí,
- *Rozšířené nástroje* – tato sada nástrojů pomáhá držet aplikaci v kvalitním stavu, obsahuje prostředky pro monitorování a testování už hotové aplikace.

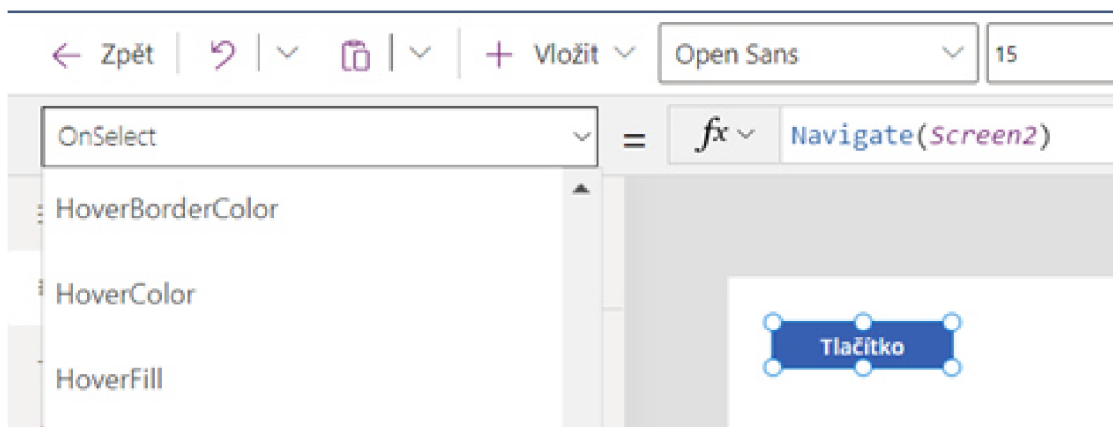
Je velmi důležitá, protože s růstem funkcionality Softwaru, bude velmi těžké, všechno testovat ručně, nebo sledovat výkonnost aplikace,

- *Hledat* – nejjednodušší nástroj, který pomůže v hledání libovolných součástí aplikace.



Obr. 10 Ruzné funkce Power Apps

Každý element, jaký je používán v aplikaci, obsahuje sadu atributů, které jsou odpovědné za jejich vlastnosti. Například nejjednodušší prvek *Tlačítko* má čtyřicet čtyři atributů. Jednou z důležitých vlastností je *OnSelect*, ostatní atributy odpovídají za barvu, text, viditelnost, režimy zobrazení, polohu v prostoru atd. Veškeré vlastnosti jsou důležité a všechny budou použité při vývoji, aby aplikace pracovala tak, jak by měla. *OnSelect* je atributem, který bude nutný, aby v něm napsaný algoritmus při stlačení tlačítka spustil funkci. Například, napsaná funkce *Navigate(Screen2)*, po dotyku *Tlačítka1* povolí přesměrování zobrazení na jinou obrazovku, než ta, na které se nachází uživatel (viz Obr.11).

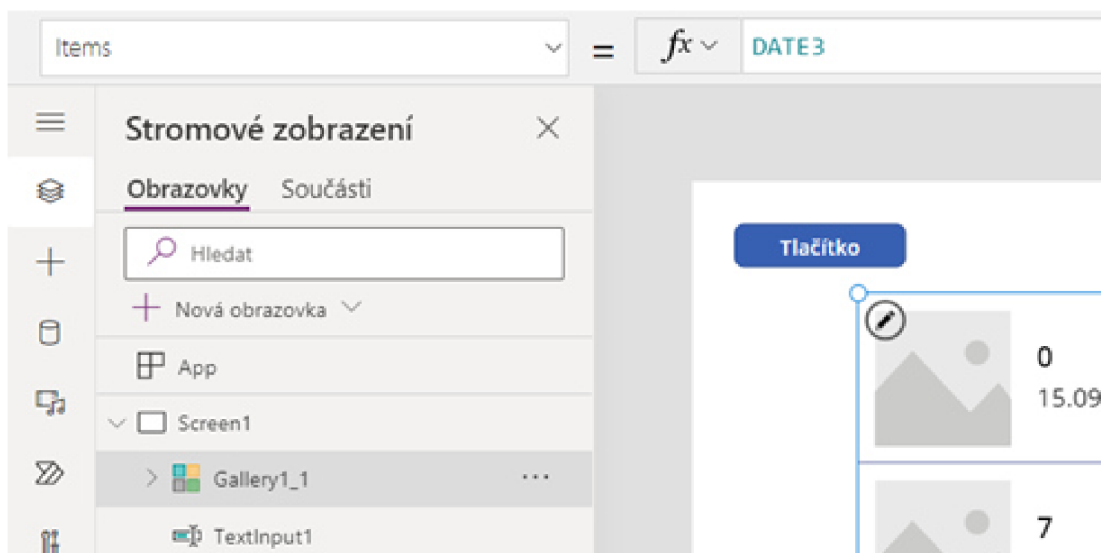


Obr. 11 Příklad elementu, jeho atributu a příkazui

V prostředí Power Apps se používají různé elementy, jako formuláře, galerie, políčka pro textové zadání, rozevřací seznamy, posuvníky, časovače a mnoho jiných. Proto se bude při vývoji často používat kolekce a proměnné. To jsou sady dat nebo jednotlivé položky s daty, které se mohou užívat přímo v prostředí Power Apps. Ta funkcionality je často používaná pro zlepšení výkonu aplikace. Například kdyby bylo potřeba se dostat do dat s identifikátorem „2“ ve sloupci „Výsledek“ a na něco se tam podívat nebo něco změnit, a zároveň by se datová obsahovala 1500 položek, tak kdyby probíhala filtrace tak velkého množství dat, určitě to by silně ovlivnilo výkon. Kdyby se však veškeré položky s identifikátorem „2“, kterých je 100, shromáždily do jedné skupiny, a pak už s ní pracovalo, tak výkon a rychlost aplikace bude mnohem lepší.

2.3.1 Spojení sady dat s galerií

Po připojení datové sady k vývojovému prostředí, je možnost jeho spojení s elementem *Galerie*, to lze provést pomocí panelu vpravo nebo ručně zapsáním kódu do atributu *Items*, jaký bude jenom jménem datového seznamu (viz Obr.12).



Obr. 12 Element Gallery a atribut Items

V následujících krocích dojde k velkému počtu změn v kódu, aby bylo možné poskytnout funkci vyhledávání nebo třídění. Při vložení a spojení galerie s daty, budou vygenerovány textové popisky, které ukážou základní informaci o položce pomocí napsané funkce *ThisItem* „*Název sloupce*“ v atributu *Text* daného elementu. Textových popisků může být veliké množství, od zadávané informace až do vygenerované SharePointem. Ale většinou se používají informace, která pomáhá identifikovat položku uživatelů.

3 Vlastní aplikace

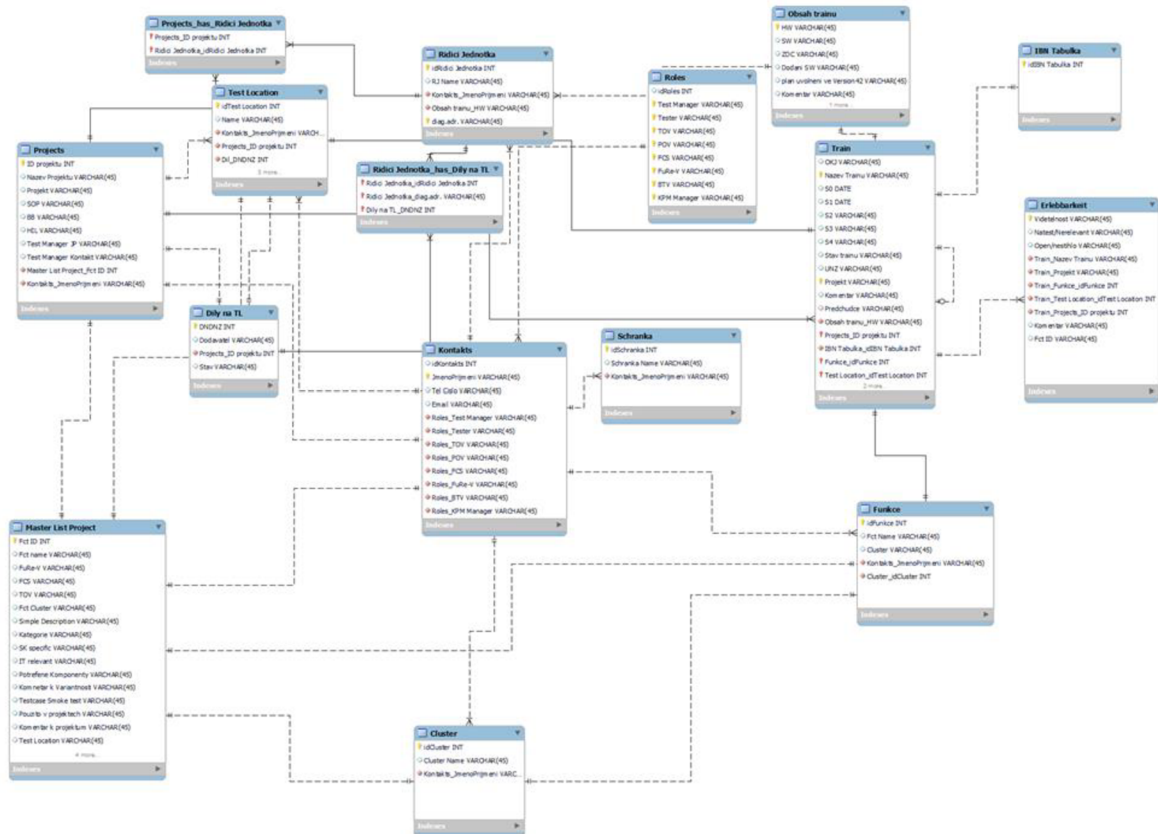
Tento odstavec představuje práci nad projektem, který byl požadován oddělením EEI ve společnosti ŠKODA AUTO a.s. Budou popsány veškeré procesy, které byly prováděny do etapy realizace, jako výběr pracovního přístupu, sestavení E-R diagramu. Kapitola tudíž bude seznamovat s technickým řešením vývoje, od začátku až do konce realizace aplikace.

3.1 Návrh aplikace

Při návrhu aplikace byl úkol opakovat funkcionalitu aplikace TIMON pomocí Power Platformy. Byly naznačeny schůzky pro organizaci a strukturování pracovního procesu. Na schůzkách bylo uvedeno, že hlavním systémem pro uložení dat a vytváření logické struktury budoucí databáze je SharePoint, z důvodů toho, že licence Microsoft 365, jehož součástí je SharePoint, už byla zakoupená. Azure SQL Database nebyl použit, kvůli tomu, že licence by musela být vyžádána u IT oddělení, což by vedlo k doplňujícím výdajům, a ztrátě času.

Při organizačním procesu byla vybrána agilní metoda, z důvodu její lepší efektivity a prokázání výsledku už na prvních etapách vývoje. Byl označen každotýdenní termín pro ukázkou udělané práce tak, že byl přiřazen technický specialista, který byl zodpovědný za projekt TIMON, u kterého bylo možné se dozvědět nutnou informaci o funkcích staré aplikace a soubory s daty pro postupné naplňování databází.

Na začátku realizace databázového systému byl navrhnut E-R diagram (viz Obr. 13), který ukazuje, jak vypadala struktura databáze nutná pro správnou práci aplikace a jejich funkcionalitu. První verze E-R diagramu byla udělána pomocí návrhářského nástroje MySQL Workbench, obsahovala veškeré typy dat, které musely být použity v procesu výroby. Dále schémata byla upravována až do konečné verze, která byla použita pro lepší pochopení účasti jednotlivých tabulek v různých aplikacích, které by na konci tvořily komplexní systém manipulace s daty.



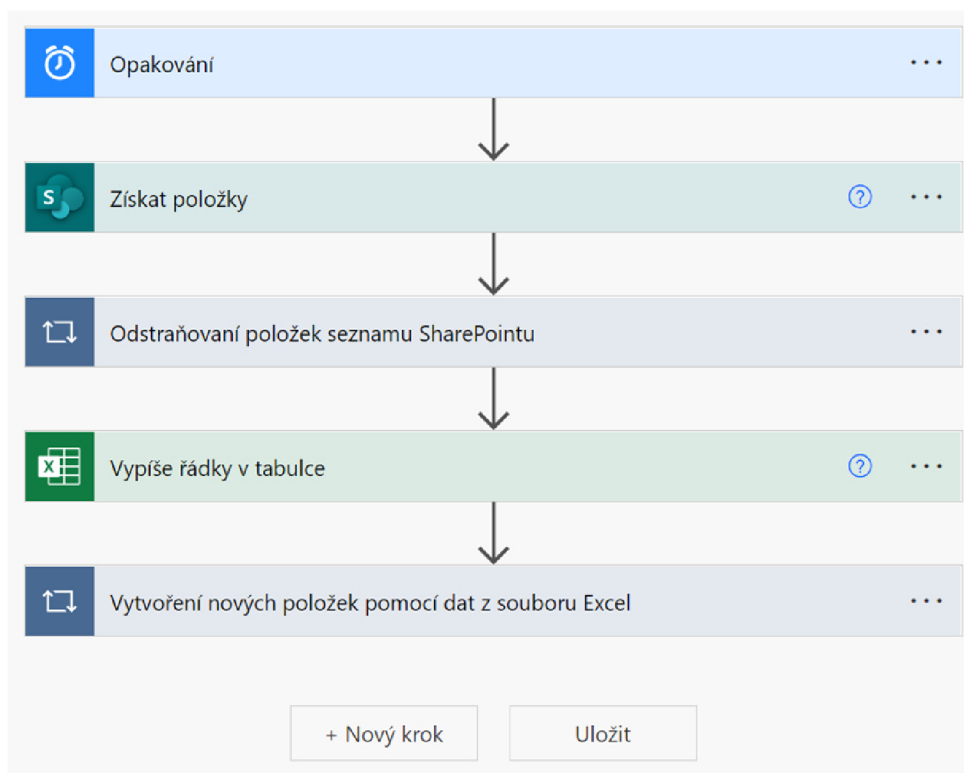
Obr. 13 E-R diagram Systému aplikaci

3.2 Realizace aplikace I

V této části bakalářské práce bude demonstrováno první řešení realizace části aplikačního systému, které bylo vyvinuto podle E-R diagramu, a požadavku od zaměstnanců na oddělení. Řešení ukáže jednotlivé části vývoje programu pomocí Microsoft Power Platform.

3.2.1 Automatizace periodické změny dat

Pro automatickou změnu dat, které se musely ukazovat pracovní týdny a datумы v aplikaci, byly použity nástroje Excel, Power Automate a SharePoint. V této části byl Excel použit pro každodenní užívání aktuálního datumu k spočítání celoročního koloběhu pracovních termínů. SharePoint sloužil, jako cloudové úložiště souborů, takže tvořil místo pro uložení nutných dat. Power Automate byl nastaven tak, aby každé pondělí v 00:00 hodin byla provedena změna dat v datovém seznamu, který by byl spojen s aplikací. Pro každotýdenní obnovení dat v Power Automate, byl použit opakující *trigger* odloženého času, součásti pro obdržení dat, cykly pro jejich smazání a další zavedení nových (viz Obr. 14).



Obr. 14 Automatické obnovení dat v čase

3.2.2 Programování a algoritmizace požadovaných funkcí

V části programování a algoritmizace budou popsány části zdrojových kódů, funkcionality Power Apps (která je spojena s vývojem aplikace), jakož i zajišťování jednotlivých funkcí s Power Automate pro zajištění správné funkcionality aplikace. Tím bude ukázán vývojový proces od začátku až do konce.

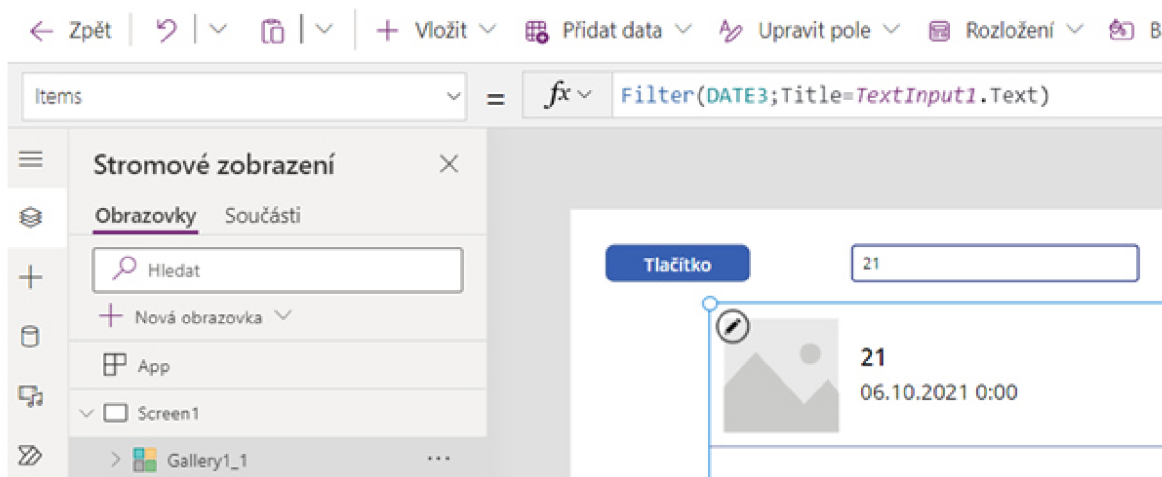
V první aplikaci, která byla zkonstruována na základě E-R diagramu spojených s požadavky uživatelů, byla aplikace Train APP. Tato aplikace pomáhá sledovat a manipulovat s daty spojené s vozy. První důležitou věcí bylo, udělat seznamy dat na SharePoint, které bylo poté možné spojit s galerií na základní obrazovce. Seznam můžeme udělat dvěma různými způsoby nebo manuálně, nebo při integraci Excel souboru. V tomto případě byla použita varianta s integrací změněného souboru.

Funkce filtrování a třídění

Pro zajištění funkce hledání, je nutné vybrat element, který bude obsahovat informaci, která bude používána, jako vstup pro funkce. To může být seznam s daty, položka textového zadání, posuvník, vypínače a všechno, co obsahuje atribut

se vstupní informací. V dané situaci se bude používat textové zadání, aby bylo jednodušší hledat položky.

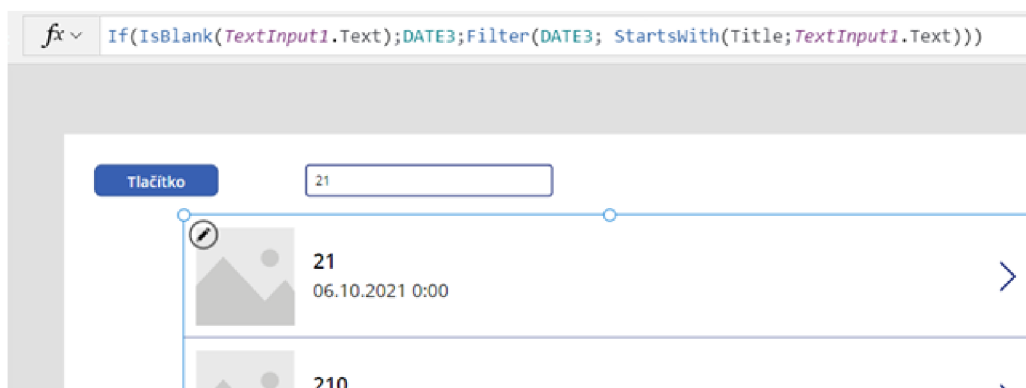
Po vložení nutného elementu, do atributu *Items* v elementu *Galerii*, musí být napsáno *Filter* (“Název datové sady”; “Název sloupce”=*TextInput.Text*). Ta funkce bude hledat položky, ve kterých bude informace odpovídající datem v hledaném sloupci stejná, jako v elementu textového zadání (viz Obr. 15).



Obr. 15 Příklad příkazu filtrace dat v Gallery

V této fázi vznikne problém: při prázdném řádku vstupního elementu nebude v galerii ukázaná žádná položka, jakož i kdyby informace byla neúplná. Proto je nutné provést modifikaci kódu pomocí podmínky a provést logický test, který povolí hledání položky, která bude začínat textem ze vstupního elementu.

V tomto případě logický test *IsBlank* ověřuje je-li zadána nějaká informace v elementu *TextInput1*. Když je vstup prázdný, tak to splňuje podmínku testu a Galerie pracuje s hodnotou *true*, což jsou data celého seznamu, v opačném případě element bude používat hodnotu *else* a filtrovat datovou sadu, což ukáže jenom hodnoty, u kterých informace v sloupci *Title* budou začínat na 21 (viz Obr. 16).



Obr. 16 Algoritmus filtrace s podmínkou

Dál bude ukázáno, jak může vypadat filtrování, které bude zajištěno pomocí low-kódu, se zajištěním dalších příkazů, jako *Sort* a *Concat*. První příkaz třídí datové sady podle ustanoveného algoritmu. Druhý příkaz rozděljuje položky vstupního seznamu dat na jednotlivé části, podle kterých je možné provádět filtrování dat v galerii. Pomocí správného používání těchto příkazů a podmínek, je zajištěn algoritmus, který je používán ve funkci filtrování dat v aplikaci (viz Obr. 17).

```
If(VarFilterTurn= true ;If(IsBlank(ComboBox1.Selected);If(TriditCombo.Selected.Value="Změněno od nejstaršího";Sort(Train;Změněno;Ascending);If(TriditCombo.Selected.Value="Změněno od nejnovějšího";Sort(Train;Změněno;Descending);If(TriditCombo.Selected.Value="Vytvořeno od nejstaršího";Sort(Train;Vytvořeno;Ascending);If(TriditCombo.Selected.Value="Vytvořeno od nejnovějšího";Sort(Train;Vytvořeno;Descending)))));Filter(If(TriditCombo.Selected.Value="Změněno od nejstaršího";Sort(Train;Změněno;Ascending);If(TriditCombo.Selected.Value="Změněno od nejnovějšího";Sort(Train;Změněno;Descending);If(TriditCombo.Selected.Value="Vytvořeno od nejstaršího";Sort(Train;Vytvořeno;Ascending);If(TriditCombo.Selected.Value="Vytvořeno od nejnovějšího";Sort(Train;Vytvořeno;Descending)))));Project in Concat(ComboBox1.SelectedItems;Název&"");If(TriditCombo.Selected.Value="Změněno od nejstaršího";Sort(Train;Změněno;Ascending);If(TriditCombo.Selected.Value="Změněno od nejnovějšího";Sort(Train;Změněno;Descending);If(TriditCombo.Selected.Value="Vytvořeno od nejstaršího";Sort(Train;Vytvořeno;Ascending);If(TriditCombo.Selected.Value="Vytvořeno od nejnovějšího";Sort(Train;Vytvořeno;Descending))))))
```

Obr. 17 Celý algoritmus filtrování napsaný pomocí low-kódu

Zlepšena funkce příloh

Power Apps už mají svou vlastní funkci, která zajišťuje uložení příloh na SharePoint pomocí formulářů. Problém ale nastává tehdy, kdy se soubor otevírá. Soubor s daty bude stáhnut do počítače, pak je možné ho opravit a poslat do aplikace znovu. Tato metoda není vhodná tehdy, jestliže chce zaměstnanec aktualizovat data v souboru přímo v prohlížeči, a neztrácet čas se zbytečnými postupy.

Pro vývoj následující funkce aplikace, byl použit Power Automate jako poskytovatel toku pro spojení části funkcionality vyvíjeného nástroje se SharePointem pro odstraňování, uložení a obdržení výstupní informace o souboru, pro další používání. V modelu je používán převod vstupního souboru do formátu *JSON* (JavaScript Object Notation). Syntaxe *JSON* je převzata ze syntaxe objektového zápisu jazyka JavaScript, ale formát je pouze textový. Kód pro čtení a generování dat tohoto formátu lze napsat v libovolném programovacím jazyce (What is JSON, 2022).

Hlavní element, který je používán pro nahrávání nutného souboru z počítače je "Přidat obrázek". Na začátku byl daný prvek určen pro nahrávání mediálních souborů do Power Apps, ale ten element může být použit i pro nahrávání souboru libovolného formátu. Prvek obsahuje dva elementy, které jsou jeho součástí:

- *AddMediaButton* – odpovědný za nahrávání souboru z počítače,
- *UploadedImage* –součástka zobrazuje nahrávaný soubor.

Jeden z nich zahrnuje kód a druhý je používán, jako vstup pro změnu formátu. Kód obsahuje proměnné a kolekce, do kterých je zapsána informace vstupních nebo výstupních dat. Hlavním atributem, který bude obsahovat algoritmus, bude *OnChange*. Ten parametr elementu zpustí děj v ten moment, kdy uživatel provede změnu v ovládacím prvku.

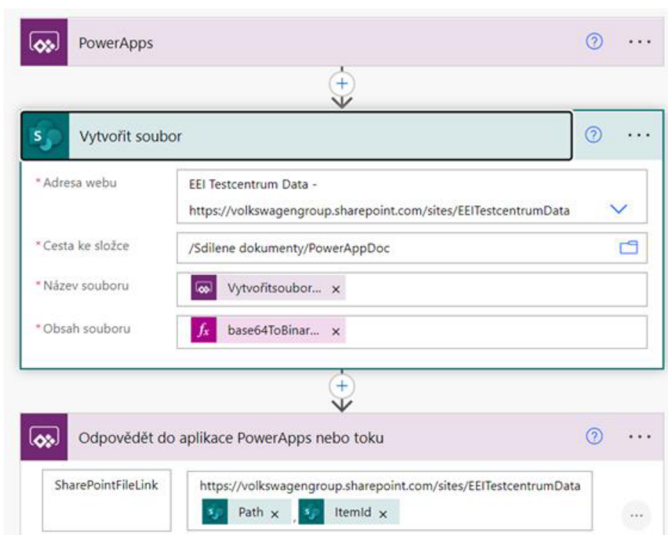
Stejně jako ve funkci filtrování a třídění, budou použité příkazy, pomocí kterých program vyplňuje algoritmy pro dosažení nutných výsledků. V tomto případě příkazy vypadají takto:

- *JSON* – převádí daty ve formát JSON,
- *Set* – vytvoří proměnnou a přiřazuje k ní data,
- *Mid* – vrátí střední znaky řetězce,
- *Len* – spočítá množství znaku v textu,
- *Find* – vrátí počáteční pozice řetězce, který byl nalezen,
- *Collect* – dodává záznamy do datové sady, obvyčejně používá pro interní kolekce,
Patch – ten příkaz dodává nebo modifikuje záznamy, může byt použit pro externí nebo interní datové sady.

Na konci pomocí výše popsaných příkazů byl napsán algoritmus, který umožňuje integrace dat do JSON formátu a posílá data do Power Automate, aby je přeformátoval do souboru, a vložil jej do datové schránky SharePointu (viz Obr. 18). Pro spojení toku Power Automate s Power Apps musí být tok na začátku přidán do vývojového prostředí. Dál je možnost je jej použít jako ještě jeden příkaz, který potřebuje vstupní data pro používání, a dál vrací výstupní hodnotu. V případě této funkce jméno toku je *SaveTheFile* s příkazem *Run*, který spustí tok (viz Obr. 19).

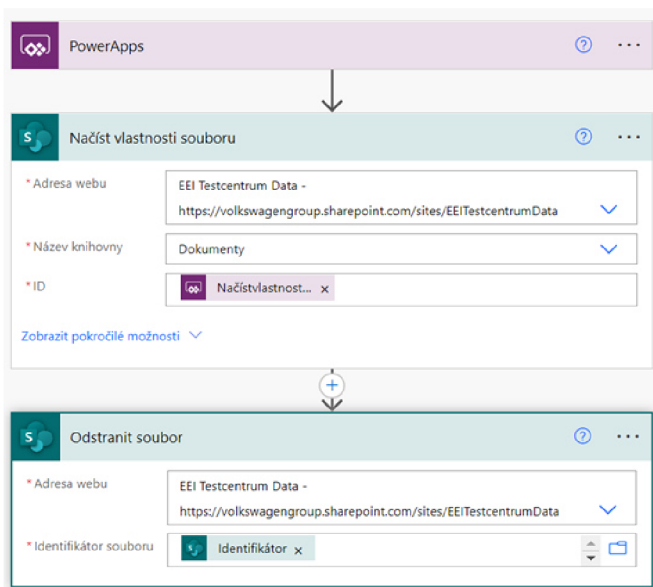
```
Set(AttachTheJSON;JSON(UploadedImage3_3.Image;IncludeBinaryData));Set(AttachBase;Mid(AttachTheJSON;Find(", ";AttachTheJSON)+1;Len(AttachTheJSON)-Find(", ";AttachTheJSON)-1));Set(AttachFileLinkId;SaveTheFile.Run(AddMediaButton3_3.FileName;AttachBase).sharepointfilelink);Set(AttachId;Mid(AttachFileLinkId;Find(", ";AttachFileLinkId)+1;Len(AttachFileLinkId)-Find(", ";AttachFileLinkId)));Set(AttachLink;Mid(AttachFileLinkId;1;Len(AttachFileLinkId)-Len(AttachId)-1));Set(AttFileList;{Title:AddMediaButton3_3.FileName;Link:AttachLink;Identifier:AttachId;IDTrain:Text(VarID.ID)});Collect(AttFiles;{Title:AddMediaButton3_3.FileName;Link:AttachLink;Identifier:AttachId;IDTrain:Text(VarID.ID)});Patch(AttachmentFilelinks;AttFileList)
```

Obr. 18 Algoritmus uložení souboru do datové schránky SharePointu pomocí JSON



Obr. 19 Tok SaveTheFile v Power Automate

Výstupem tohoto toku bude odkaz na nový soubor v knihovně SharePointu a jeho identifikátor, který bude použit v podfunkci odstraňování (viz Obr. 20). Odkaz a identifikátor budou následně přidány do seznamu v SharePointu, aby vždy byl přístup k datům, které jsou přidány pomocí aplikace. Je tedy nutné zajistit možnost zobrazení jenom těch souborů, které odpovídají vybranému Trainu a musí do vytvořeného seznamu pro tyto záznamy přidat i sloupec *TrainID*. Pro spuštění souboru přímo z aplikace byl použit příkaz *Launch*.



Obr. 20 Funkce Odstraňování souboru ze Sharepointu

Funkcionalita GridView

Funkcionalita *GridView* byla zajištěna pro snadné manipulace s daty v SharePointových seznamech přímo v elementu Galerii proto, aby nemusel uživatel otevírat jednu položku a zaměřovat záznam dat z formuláře. Bylo to uděláno z důvodů, že je přehlednější dívat na celý seznam dat zapsaných do Trainu a zajišťovat změny bez ztráty času. V algoritmu byly použity příkazy *Patch*, *Collect*, *ClearCollect*, *Set*, *Remove*, *Select*, *UpdateIf* a *UpdateContext*.

Remove je příkazem, který odstraňuje vybranou položku z datové sady. *Select* vybírá danou položku, pro další pracování s ní. *UpdateIf* je podmínkovým příkazem, který modifikuje nebo přidává záznamy, při splnění specifické podmínky. *UpdateContext* obnovuje kontext proměnné na nový. *ClearCollect* odstraňuje veškeré záznamy kolekce a dále přidává nové řetězce do stejné kolekce.

Pro splnění funkcionality systému byli použité různé prvky a jejich atributy, jako galerie a její atributy *OnSelect* a *OnChange*. Do systému byli zařazené takové elementy, jako Tlačítka, Rozevírací seznamy, Textové zadání a Přepínací tlačítka. Práce této funkce byla zajištěna pomocí algoritmu, který přidává nebo modifikuje data v kolekci, a pak ukládá změny do SharePointu. Pro odstraňování položek systém umísťuje ID jednotlivých záznamů v kolekci, následně bude seznam položek odstraněn z webového seznamu.

```
If(VarDridEdit;Patch('Train Components';UpdateIf(TrainRJComponents;And(ID=ThisRecord.ID;Vytvořeno=Blank());{ID:Blank();DiagAdr:DiagAdr;RidiciJednotka:Title;CisloDilu:CisloDilu;HV:HV;SF:SF;HW:HW;SW:SW;DodaniSW:DodaniSW;PlanUvolneniVeVersion42:PlanUvolneniVeVersion42;ZDCVersion:ZDCVersion;ZDCDetailID:ZDCDetailID;'HV-QT':'HV-QT';Změna:Změna;IDTRAIN:Gallery2.Selected.ID}));Remove('Train Components';ColDelete);;Set(VarDridEdit;!VarDridEdit);;Select(ButtonLoadData);Set(VarDridEdit;!VarDridEdit);;Select(ButtonLoadData)
```

Obr. 21 Algoritmus umožňující práci s daty v přímo v elementu Gallery

3.2.3 Realizace UI

Jakmile je dosaženo požadované funkčnosti programu, je nezbytným krokem aktualizace UI pro snadnější a plynulejší práci s daty pro uživatele. V tomto případě je nutno zajistit nejvhodnější rozložení elementu na jednotlivých obrazovkách, aby program byl intuitivní. *Formuláře*, které jsou elementy Power Apps nezahrnují funkci libovolného pohybu vnitřních položek, ale dávají představu, jaké elementy jsou potřebné pro upravený formulář. *Formuláře* automaticky načtou všechny názvy sloupců při připojení datových sad, a vygenerují buňky s elementy pro modifikaci nebo zadávání dat. Pokud je zadávání informací intuitivní i bez nutnosti speciálního

rozložení, formulář lze použít i ve finální verzi programu. Pro realizaci UI byly v některých elementech použity atributy *Visible* a *DisplayView*. První se používá ke kontrole viditelnosti elementu, druhý pro jeho zobrazení.

V dalším vývoji UI se používají takzvané kontejnery, které mohou obsahovat sadu ovládacích prvků, mají své vlastní vlastnosti, pomáhají rozložit sady souvisejících prvků a dále manipulovat s nimi, jako s jedním objektem. Jsou velmi užitečné v realizaci UI. Nezbytným elementem správného UI je design. Z důvodu toho že program byl vyvinut pro ŠKODU AUTO a.s. Hlavními barvami budou bílá a zelená. Obrázky kompletního programu jsou v Příloze č. 2.

3.3 Realizace vizualizací pomocí Power BI

Realizaci vizualizací v Power BI probíhá většinou v „těžké“ aplikaci, tj. přímo na počítači. To je z toho důvodu, že Power BI povoluje velké změny jenom v desktopovém programu. Po ukončení nastavení vizualizace, je možnost nahrát webové stránky s funkcionalitou, která byla zajištěna na počítači. Při práci s Power BI je nutnost připojit datové sady, s kterými probíhala práce v aplikaci. Teď už při libovolné změně jednotlivých záznamů ve vyvinutém softwaru se budou objevovat v Power BI.

V tomto systému je velké množství funkcí, takových jako libovolné možnosti vizualizací (Grafy, tabulky atd.) dále také možnosti změny jejich formátu. Z toho důvodu je platforma velmi užitečná pro tvorbu povedených reportů v reálném čase. Existuje funkce udělování svého modelu, tj. možnost spojení dat přes stejné položky. Pro práci není potřeba znát žádný programovací jazyk, funkcionalita pro filtrování a třídění, už je vložena v elementech, které je možné přidávat přímo pro jednotlivé datové sady nebo několik datových sad. Obrázky vizualizací dat z aplikací budou v Příloze č. 3.

3.4 Výsledky realizace

Výsledkem práce se stala aplikace, která umožňovala, manipulovat s řídicími jednotkami pro vozy a zavádět nové projekty, jako stávající aplikace. Vyvinutý program má pokročilou funkci příloh, kterou neměl stávající program. Veškerá data byla uložena na webových stránkách SharePointu, což umožnilo jednodušší systematizaci výsledku a jejich analýzu. Hlavním problémem se však stala

výkonnost aplikace a přístup k ní. Společnost neměla Power Platform Premium, což nepovolovalo přístup k aplikaci více než pěti lidem.

Na konci realizace aplikace a jeho zavedení do práce bylo zjištěno pomocí zpětné vazby, že systém nebyl výkonnější než stávající aplikace, ale poskytoval lepší vizualizaci dat. Bylo rozhodnuto, že práce bude stále probíhat ve stávajícím systému a Microsoft Power Platform bude používán pro vývoj jednodušších aplikací, ve kterých nebudou používány velké sady dat, a také pro vizualizaci reportu a automatizaci kancelářských procesů.

4 Možnosti dalšího využití Power Platform v rámci ŠKODA AUTO a.s

Power Platforma je komplexním systémem, a možnosti jejího využití jsou prakticky nekonečné, což umožňuje používat toto konstrukční prostředí v mnoha oblastech v rámci společnosti. Například v docházkovém systému, účtovacím systému, oblastech, kde je potřeba práce s daty, automatizace různých procesů, jako automatické sčítání dat a jejich přenos na webové stránky, kde to bude představeno ve formě reportu atd. Jako příklad může sloužit takový systém, při kterém mail od dodavatelů s přílohou ohledně dodávání zásob, spustí algoritmus, díky čemuž příloha s daty, ve které jednotlivé sloupce budou automaticky přepočítané na nutné hodnoty a zařazené do databází společnosti s upozorněním odpovědných zaměstnanců.

Power Apps může být například použit ve vývoji aplikace evidence jednotek s používáním multimediálních souborů a rozpoznávání QR-kódů, aplikace, ve které bude možnost podávat žádosti pro digitální podpis, nebo pro interní aplikaci počítání benefitů. Velkým plusem je to, že Power Automate obsahuje velké množství šablon, nebo už hotových aplikací, které jsou prací komunity Power Platform a to se týká i Automate a BI.

Power BI je možné používat pro vizualizaci statistických výsledků v grafické podobě, tvoření reportů a jejich prezentací, tak že je možné je používat, jako návrhářský nástroj pro aplikace, nebo pro tvoření interaktivních webových stránek. Nástroj bude velkou pomocí na odděleních obchodu nebo statistických odděleních.

Power Automate zajišťuje široké možnosti pro automatizaci pracovních procesů, jako automatické odpovědi na emaily, upozornění, periodické logování dat, automatické načítání a ukládání dat, nebo jejich modifikace, uložení příloh, automatické spuštění analýzy, vytvoření úkolu v Planetu atd.

Závěr

Cíle bakalářské práce byly následující: seznámení s teoretickými východisky, se současným stavem softwaru, jakož i s popisem konstrukčního prostředí Microsoft Power Platform a vývoji aplikace na jeho základě.

V první části prací byla popsána částečná problematika vývoje aplikace v IT oblasti. Je to opravdu velká oblast, a proto byla popsána jenom ta část, která byla používaná ve vývoji projektu. Byly představeny zásady návrhu aplikace, analýza procesu a projektu, zásady vývoje, teoretické znalosti o databázích a datových sadách, představení analýzy vývojového prostředí Microsoft Power Platform a její jednotlivé součásti. V následující části bakalářské práce byl představen současný stav softwaru na oddělení a její funkcionality. Dál byl představen návrh vlastního řešení a jeho realizace pomocí nástrojů platformy a navazujícího databázového systému. Hlavními informacemi v této práci byla široká funkcionality Microsoft Power Platform a její praktické používání při vývoji aplikace, a další možnosti jejího využití v rámci ŠKODA AUTO a.s.

Na konci, při shromažďování veškerých informací ohledně projektu je nutnost říct, že Power Platform je velmi užitečným nástrojem, který by mohl zlepšit výkonnost a efektivitu na pracovišti. Obsahuje pět různých nástrojů, které by mohly pomoci v různých oblastech práce. Platforma se bude s časem rozvíjet a optimalizovat a bude široce používána ve veškerých oblastech byznysu.

Seznam literatury

BEDNAŘ, Vojtěch. Zavádění informačních systémů a motivace personálu. *BusinessINFO.cz* [online]. 2022, 03. 01. 2022, 8 [cit. 2022-12-02]. Dostupné z: <https://www.businessinfo.cz/clanky/zavadeni-informacnich-systemu-a-motivace-personalu/>

BERGER, Roman. Business aplikace do mobilu a tabletu propojená se CRM a ERP. *Www.konicaminoltaits.cz* [online]. Praha: Konica Minolta IT Solutions Czech, 2020, 15. 9. 2020 [cit. 2022-11-24]. Dostupné z: <https://www.konicaminoltaits.cz/dynamics365/business-aplikace-do-mobilu-a-tabletu-propojena-se-crm-a-erp/>

BERGER, Roman a Jan HLAVINKA. Analýza podnikových dat ze CRM, ERP i jiných zdrojů díky Power. *Www.konicaminoltaits.cz* [online]. Praha: Konica Minolta IT Solutions Czech, 2020, 17. 9. 2020 [cit. 2022-11-24]. Dostupné z: <https://www.konicaminoltaits.cz/dynamics365/analyza-podnikovych-dat-ze-crm-erp-i-jinych-zdroju/>

Co je Power Pages?. *Learn.microsoft.com* [online]. Redmond: Microsoft, 2022, 14. 10. 2022 [cit. 2022-11-29]. Dostupné z: <https://learn.microsoft.com/cs-cz/power-pages/introduction>

Co je Azure SQL Database?. *Learn.microsoft.com* [online]. Redmond: Microsoft, 2022, 29. 09. 2022 [cit. 2022-11-29]. Dostupné z: <https://learn.microsoft.com/cs-cz/azure/azure-sql/database/sql-database-paas-overview?view=azuresql>

DOLEŽAL, Jan, Pavel MÁČHAL a Branislav LACKO. *Projektový management podle IPMA. 2., aktualiz. a dopl. vyd.* Praha: Grada, 2012. Expert (Grada). ISBN 8024742756.

JONES, Justin a Scott WADDELL. The Cascading Costs of Waterfall. *Medium.com* [online]. San Francisco: Medium, 2019, 23. 04. 2019 [cit. 2022-11-29]. Dostupné z: <https://medium.com/@joneswaddell/the-cascading-costs-of-waterfall-5c3b1b8beaec>

INSTITUTE, Project Management. *A guide to the project management body of knowledge*. Sixth. vyd. Newtown Square: Project Management Institute, 2017. ISBN 9781628251845;9781628253825;1628253827;1628251840;

Magický trojúhelník projektového řízení. *Managementmania.com* [online]. Tursko: Symphera, 2019, 05. 05. 2019 [cit. 2022-11-29]. Dostupné z: <https://managementmania.com/cs/magicky-trojuhelnik-projektoveho-rizeni>

MAREK, Jiří. Power Platform – sametová revoluce ve vývoji business aplikací. *Www.konicaminoltaits.cz* [online]. Praha: Konica Minolta IT Solutions Czech, 2020, 18. 9. 2020 [cit. 2022-11-24]. Dostupné z: <https://www.konicaminoltaits.cz/dynamics365/power-platform-sametova-revoluce-ve-vyvoji-business-aplikaci/>

Microsoft Excel. *it.muni.cz* [online]. Brno: Masarykova univerzita, 2022 [cit. 2022-11-29]. Dostupné z: <https://it.muni.cz/sluzby/microsoft-excel#:~:text=MS%20Excel%20je%20tabulkov%C3%BD%20procesor,slo%C5%BEit%C4%9Bj%C5%A1%C3%AD%20%C4%8Di%20automatizovan%C3%A9%20v%C3%BDpo%C4%8Dtov%C3%A9%20aktivty>.

BAIZINI, Giacomo. PowerApps 101: where do I store my data?. *Baizini-it.com* [online]. Milan: Giacomo Baizini's business IT blog, 2017, 29. 08. 2017 [cit. 2022-11-29]. Dostupné z: <https://baizini-it.com/blog/index.php/2017/08/29/powerapps-101-data-storage/>

PETROV, Alex. *Database internals: a deep dive into how distributed data systems work*. First. vyd. Beijing;Farnham;Sebastopol;Tokyo;Boston,; O'Reilly, 2019. ISBN 9781492040347;1492040347;

Procesní analýza (Process analysis). In: *ManagementMania.com* [online]. Wilmington (DE) 2011-2022, 05.09.2018 [cit. 25.11.2022]. Dostupné z: <https://managementmania.com/cs/analyza-procesu-procesni-analyza>

Přehled Power Virtual Agents. *Learn.microsoft.com* [online]. Redmond: Microsoft, 2022, 28. 11. 2022 [cit. 2022-11-29]. Dostupné z: <https://learn.microsoft.com/cs-cz/power-virtual-agents/fundamentals-what-is-power-virtual-agents>

RATHEE, Kovid. 7 Tips for a Good ER Diagram Layout. *Vertabelo.com* [online]. Melbourne: Servian, 2021 [cit. 2022-11-29]. Dostupné z: <https://vertabelo.com/blog/vertabelo-tips-good-er-diagram-layout/>

SILBERSCHATZ, Abraham, Henry F. KORTH a S. SUDARSHAN. *Database system concepts*. Seventh. vyd. New York, NY: McGraw-Hill Education, 2020. ISBN 9781260084504;1260084507;

SLAWEK-POLCZYNSKA, Alina. Is Agile always the best solution for software development projects?. *Www.soldevelo.com* [online]. Gdynia: SolDevelo, 2020, 26. 11. 2020 [cit. 2022-11-29]. Dostupné z: <https://www.soldevelo.com/blog/is-agile-always-the-best-solution-for-software-development-projects/>

STETSON, Chris. Principles of Modern Application Development. *Www.nginx.com* [online]. California: NGINX, 2018, 25. 8. 2018 [cit. 2022-11-23]. Dostupné z: <https://www.nginx.com/blog/principles-of-modern-application-development/>

What Is a Database?. *Www.oracle.com* [online]. Austin: Oracle, 2020, 08. 03. 2020 [cit. 2022-11-29]. Dostupné z: <https://www.oracle.com/cz/database/what-is-database/>

What is JSON. *Www.w3schools.com* [online]. Sandnes: Refsnes Data AS, 2022, 02. 02. 2022 [cit. 2022-11-29]. Dostupné z: https://www.w3schools.com/whatis/whatis_json.asp

Začínáme s SharePointem. *Support.microsoft.com* [online]. Redmond: Microsoft, 2022, 10. 03. 2022 [cit. 2022-11-29]. Dostupné z: <https://support.microsoft.com/cs-cz/office/za%C4%8D%C3%ADn%C3%A1me-s-sharepointem-909ec2f0-05c8-4e92-8ad3-3f8b0b6cf261>

Seznam obrázků a tabulek

Seznam obrázků

Obr. 1 Špatně řízené projekty.....	10
Obr. 2 Dobře řízený projekt	10
Obr. 3 Metoda Waterfall	13
Obr. 4 Metoda Agile	13
Obr. 5 Příklad uspořádaného E-R diagramu	17
Obr. 6 Příklad sad entit.....	18
Obr. 7 Příklad relace mezi entity.....	18
Obr. 8 Příklad sady vztahu	18
Obr. 9 Obrazovky a Součásti.....	22
Obr. 10 Ruzné funkce Powe Apps	23
Obr. 11 Příklad elementu, jeho atributu a příkážu.....	24
Obr. 12 Element Gallery a atribut Items	25
Obr. 13 E-R diagram Systému aplikaci	27
Obr. 14 Automatické obnovení dat v čase.....	28
Obr. 15 Příkaz filtrace dat v Gallery.....	29
Obr. 16 Algoritmus filtrace s podmínkou	29
Obr. 17 Celý algoritmus filtrování napsaný pomocí low-kódu.....	30
Obr. 18 Algoritmus uložení souboru do datové schránky SharePointu pomocí JSON.....	31
Obr. 19 Tok SaveTheFile v Power Automate	32
Obr. 20 Funkce Odstraňování souboru ze Sharepointu	32
Obr. 21 Algoritmus umožňující práce s daty v přímo v elementu Gallery	33
Obr. 22 Zakladní obrazovka TIMON.....	44
Obr. 23 Zadávání nové řídicí jednotky pro voz.....	44

Obr. 24 Upravení řídicí jednotky	45
Obr. 25 Základní obrazovka vyvinuté aplikaci Train APP	46
Obr. 26 Úprava trainu v Train APP	46
Obr. 27 Zadávání trainu v Train APP	47
Obr. 28 Zástavba trainu v Train APP.....	47
Obr. 29 Funkce GridView	48
Obr. 30 Zadávání nové řídicí jednotky pro voz	48
Obr. 31 Základní obrazovka v Power BI.....	49
Obr. 32 Zástavba Trainu v Power BI	49
Obr. 33 Srovnání trainu v Power BI.....	50

Seznam příloh

Příloha č. 1 Aplikace TIMON	44
Příloha č. 2 Aplikace.....	46
Příloha č. 3 BI.....	49

Příloha č. 1 Aplikace TIMON

Internal Public Internal Confidential Secret

TIMON - Požadavky Aktual KW 4/12/2012

Soubor Hlavní Informace Nový Úpravy Test Zobrazení Stanovisko Testplán Report Fokus Kontrola Výhled Překlopení SOP Evidence RJ Správa KF
 Ukončit Možnosti Podpora Účastník Vymaz/Obnov Filtrování Překresli Vyber Releaseplán Vyber Aktuální stav Vyber Export CSV Editor Editor Editor Editor

ID	Typ testu	Projekt	SOP	Derivát	Diag. adr.	Řídicí jednotka	Dodání RJ	Nový HW	HW	SW	Číslo dílu	Test. místa	Var. vozu	Typ test
2835	RJ	VOZ_A	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2836	RJ	VOZ_B	XX/YY	Pev	[XXXX]		Ano	Ne	XXX	XXXX				
2837	RJ	VOZ_C	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2838	RJ	VOZ_D	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2839	RJ	VOZ_E	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2840	RJ	VOZ_F	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2841	RJ	VOZ_A	XX/YY	Pev	[XXXX]		Ano	Ne	XXX	XXXX				
2842	RJ	VOZ_B	XX/YY	Pev	[XXXX]		Ano	Ne	XXX	XXXX				
2843	RJ	VOZ_F	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2844	RJ	VOZ_F	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2845	RJ	VOZ_C	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2846	RJ	VOZ_D	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2847	RJ	VOZ_D	XX/YY	Kov	[XXXX]		Ano	Ano	XXX	XXXX				
2848	RJ	VOZ_D	XX/YY	Kov	[XXXX]		Ano	Ano	XXX	XXXX				
2849	RJ	VOZ_E	XX/YY	Kov	[XXXX]		Ano	Ano	XXX	XXXX				
2850	RJ	VOZ_C	XX/YY	Kov	[XXXX]		Ano	Ano	XXX	XXXX				
2851	RJ	VOZ_E	XX/YY	Kov	[XXXX]		Ano	Ano	XXX	XXXX				
2852	RJ	VOZ_A	XX/YY	Kov	[XXXX]		Ano	Ano	XXX	XXXX				
2853	RJ	VOZ_A	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2854	RJ	VOZ_A	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2855	RJ	VOZ_C	XX/YY	Pev	[XXXX]		Ano	Ne	XXX	XXXX				
2856	RJ	VOZ_G	XX/YY	Pev	[XXXX]		Ano	Ne	XXX	XXXX				
2857	RJ	VOZ_A	XX/YY	Pev	[XXXX]		Ano	Ne	XXX	XXXX				
2858	RJ	VOZ_B	XX/YY	Pev	[XXXX]		Ano	Ne	XXX	XXXX				
2859	RJ	VOZ_A	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2860	RJ	VOZ_A	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				
2861	RJ	VOZ_A	XX/YY	Kov	[XXXX]		Ano	Ne	XXX	XXXX				

Obr. 22 Zakladní obrazovka TIMON

Internal Public Internal Confidential Secret

TIMON - Požadavky

Soubor Hlavní Informace Nový Úpravy Test Zobrazení Stanovisko Testplán Report Fokus Kontrola Výhled Překlopení SOP Evidence RJ Správa KF
 Ukončit Možnosti Podpora Účastník Vymaz/Obnov Filtrování Překresli Vyber Releaseplán Vyber Aktuální stav Vyber Export CSV Editor Editor Editor Editor

Nový test - Řídicí jednotka

ID: 2767, 2768, 2769, 2770, 2771, 2772, 2773, 2774, 2775, 2776, 2777, 2778, 2779, 2780

1. Projekt: SOP - Derivát
 2. Říj/diagn. adresa: Vyber
 3. HW: []
 4. Nový HW: Ano [] Ne []
 5. SW: []
 6. Číslo dílu (SW): []
 7. Varianta vozu (užší Dataset!): []
 8. Typ testu(ů): Identifikace [] Komunikace [] Klidové proudy [] Funkční test []
 9. Změření testu (změny od posledního testu): []
 10. KW testu: Vyber
 11. Nejzavazší datum uvolnění: []

Potvrzuji odpovídající naplnění systémů (System42, CP-Tool)

Prosím o dodání HW nebo SW do pátku 12:00 v týdnu před požadovaným KW testu. Pokud nebude dodáno a nebude komunikována výjimka, nebude test na danou jednotku realizován!

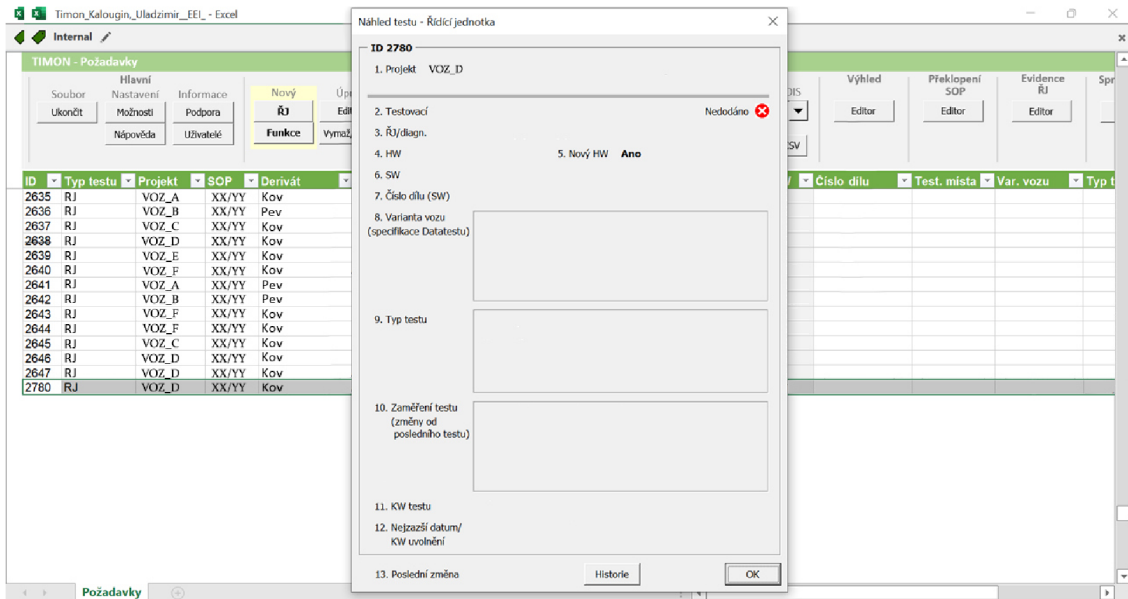
Doplňující informace: Telefon, Email, Umístění, Důly předávejte v počtu pro testovací místa. V případě potřeby testování prosím kontaktujte Testmanažera.

Překlopení SOP Editor Evidence RJ Editor Správa KF Editor

Test. místa Var. vozu Typ test

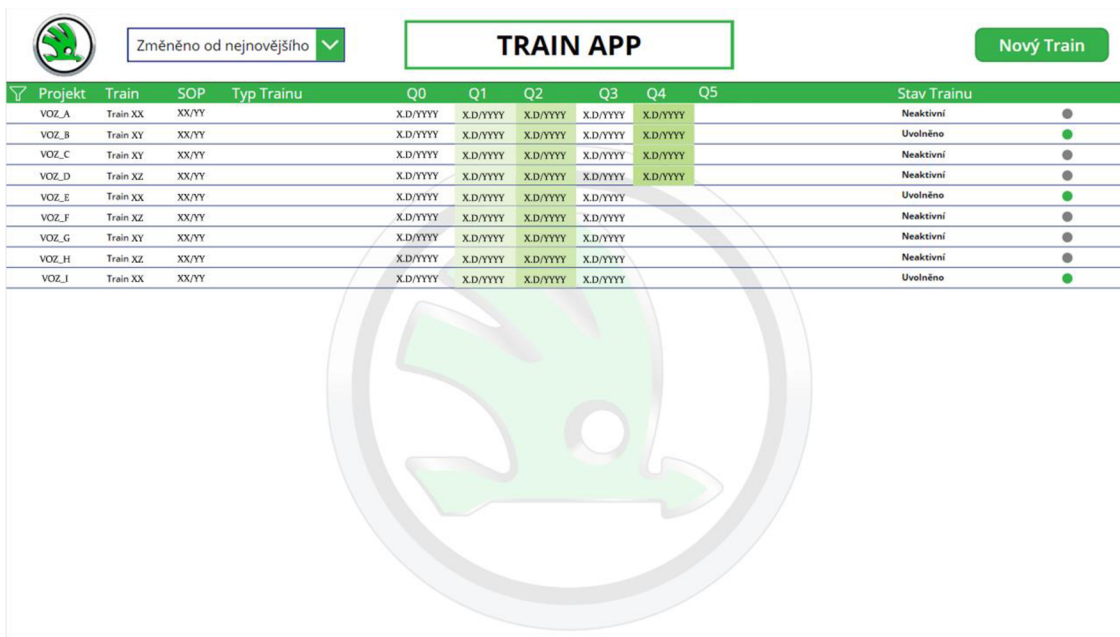
Uložit Storno

Obr. 23 Zadávání nové řídicí jednotky pro voz



Obr. 24 Upravení řídicí jednotky

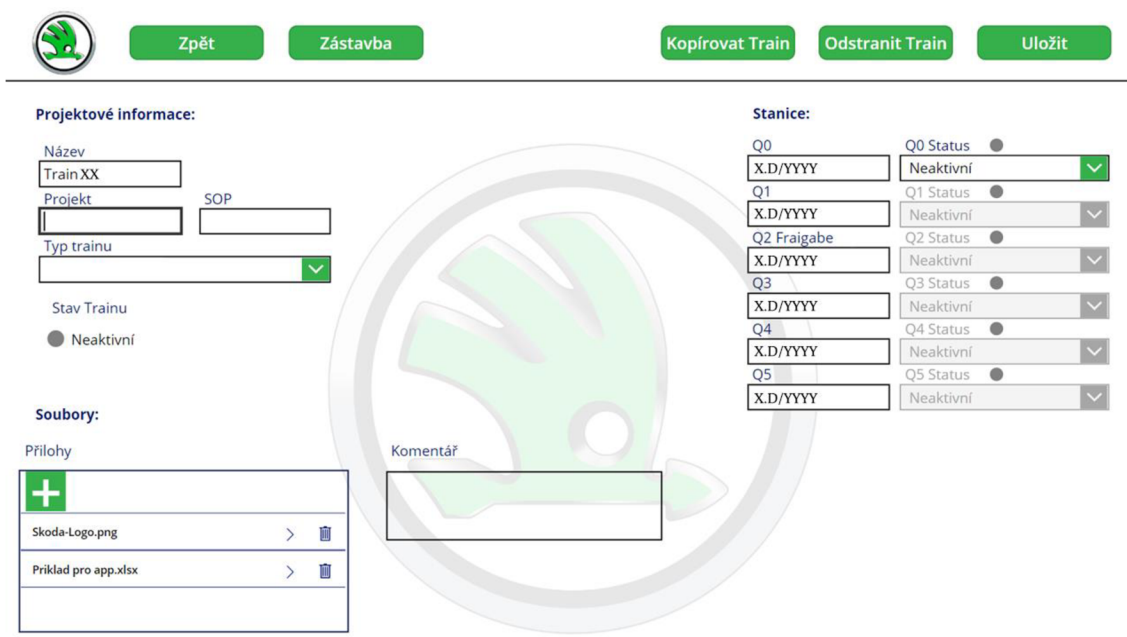
Příloha č. 2 Aplikace



The screenshot shows the main interface of the Train APP. At the top left is a logo and a dropdown menu set to 'Změněno od největšího'. In the center is a title box 'TRAIN APP'. At the top right is a green button 'Nový Train'. Below is a table with columns: Projekt, Train, SOP, Typ Trainu, Q0, Q1, Q2, Q3, Q4, Q5, and Stav Trainu. The table contains 10 rows of data. A large, faint logo is visible in the background.

Projekt	Train	SOP	Typ Trainu	Q0	Q1	Q2	Q3	Q4	Q5	Stav Trainu
VOZ_A	Train XX	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Neaktivní
VOZ_B	Train XY	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Uvolněno
VOZ_C	Train XY	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Neaktivní
VOZ_D	Train XZ	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Neaktivní
VOZ_E	Train XX	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Uvolněno
VOZ_F	Train XZ	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Neaktivní
VOZ_G	Train XY	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Neaktivní
VOZ_H	Train XZ	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Neaktivní
VOZ_J	Train XX	XX/YY		X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY	X.D/YYYY		Uvolněno

Obr. 25 Základní obrazovka vyvinuté aplikaci Train APP



The screenshot shows the 'Edit Train' form in the Train APP. At the top are buttons: 'Zpět', 'Zástavba', 'Kopírovat Train', 'Odstranit Train', and 'Uložit'. The form is divided into sections: 'Projektové informace' (Name, Project, SOP, Train Type, Status), 'Stanice' (Stations Q0-Q5 with date and status), 'Soubory' (Attachments), and 'Komentář' (Comment). A large, faint logo is visible in the background.

Projektové informace:

Název:
 Projekt: SOP:
 Typ trainu:
 Stav Trainu: Neaktivní

Stanice:

Q0: Q0 Status: Neaktivní
 Q1: Q1 Status: Neaktivní
 Q2 Fraigabe: Q2 Status: Neaktivní
 Q3: Q3 Status: Neaktivní
 Q4: Q4 Status: Neaktivní
 Q5: Q5 Status: Neaktivní

Soubory:

Přílohy:

- Skoda-Logo.png
- Příklad pro app.xlsx

Komentář:

Obr. 26 Úprava trainu v Train APP

ANOTAČNÍ ZÁZNAM

AUTOR	Uladzimir Kalougin		
STUDIJNÍ PROGRAM/OBOR/SPECIALIZACE	Podniková ekonomika a manažerská informatika		
NÁZEV PRÁCE	Návrh a vývoj aplikace v prostředí Microsoft Power Platform		
VEDOUCÍ PRÁCE	Ing. Lukáš Herout Ph. D.		
KATEDRA	KI – Katedra informatiky	ROK ODEVZDÁNÍ	2022
POČET STRAN	50		
POČET OBRÁZKŮ	33		
POČET TABULEK	0		
POČET PŘÍLOH	3		
STRUČNÝ POPIS	<p>Práce byla zaměřena na proces návrhu a vývoje aplikace v prostředí Microsoft Power Platform. Cílem bylo seznámit s problematikou vývoje aplikace, navrhnout a vyvinout aplikace na základě Power Platform a popsat další možnosti jeho užívání na pracovišti. V procesu vývoje aplikace, byly použité nástroje Power Apps, Power Automate a Power BI, databázový systém byl zajištěn pomocí SharePointu. Výsledkem bylo zjištění, že Power Platform je možné použít pro vývoj jednoduchých aplikací, automatizaci pracovních procesů a vizualizaci dat.</p>		
KLÍČOVÁ SLOVA	Microsoft Power Platform, Power Apps, Power Automate, Power BI, databázový systém, SharePoint		

ANNOTATION

AUTHOR	Uladzimir Kalougin		
FIELD	Business Informatics		
THESIS TITLE	Application design and development in Microsoft Power Platform environment		
SUPERVISOR	Ing. Lukáš Herout Ph. D.		
DEPARTMENT	KI – Department of Informatics	YEAR	2022
NUMBER OF PAGES	50		
NUMBER OF PICTURES	33		
NUMBER OF TABLES	0		
NUMBER OF APPENDICES	3		
SUMMARY	<p>The work focused on the process of designing and developing an application in the Microsoft Power Platform environment. The aim was to introduce the issues of application development, to design and develop applications based on Power Platform and to describe other possibilities of its use in the workplace. In the application development process, Power Apps, Power Automate and Power BI tools were used, the database system was provided with the help of SharePoint. As a result, the solution was that Power Platform can be used for developing simple applications, workflow automation and data visualization.</p>		
KEY WORDS	Microsoft Power Platform, Power Apps, Power Automate, Power BI, database system, SharePoint		