Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Diplomová práce

Návrh a implementace notifikační aplikace pro OS MindSphere pro společnost VDT Technology a. s.

Bc. Dominik Bureš

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Dominik Bureš

Informatika

Název práce

Návrh a implementace notifikační aplikace pro OS MindSphere pro společnost VDT Technology a. s.

Název anglicky

Design and implementation of a notification application for OS MindSphere for the company VDT Technology a. s.

Cíle práce

Cílem teoretické části diplomové práce je popsat základy vývoje aplikací a použité technologie pro vývoj notifikační aplikace formou literární rešerše. Dále provést analýzu a zhodnocení existujících notifikačních aplikací v OS MindSphere.
Cílem praktické části diplomové práce je provést analýzu uživatelských požadavků na notifikační aplikaci a následně vytvořit návrh vlastní notifikační aplikace, který bude implementován a otestován v OS MindSphere.

Metodika

Na základě studia odborných zdrojů budou popsány použité technologie pro vývoj aplikace formou literární rešerše. Bude provedena analýza uživatelských požadavků na notifikační aplikaci a popsána vhodnými nástroji (Use Case). Dále bude provedena analýza a zhodnocení existujících aplikací v OS MindSphere a následně v souladu s doporučenými postupy sofwarového inženýrství bude vytvořen návrh vlastní notifikační aplikace (konceptuální model). Návrh aplikace bude implementován v OS MindSphere a řádně otestován.

Doporučený rozsah práce

60-80 stran

Klíčová slova

notifikační aplikace, softwarový návrh, low-code, Mendix, MindSphere

---

Doporučené zdroje informací

KAVIS, Michael. ARCHITECTING THE CLOUD. Hoboken: John Wiley, 2014. ISBN 978-1-118-69177-9.

KENNEWEG, Bryan, Imran KASAM a Micah MCMULLEN. Building Low-Code Applications with Mendix. Birmingham: Packt Publishing, 2021. ISBN 978-1-80020-142-2.

PATNI, Sanjay. Pro RESTful APIs: Design, Build and Integrate with REST, JSON, XML and JAX-RS. 2017. Berkeley, CA: Apress, 2017. ISBN 978-1-4842-2664-3.

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

Ing. Dana Vynikarová, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 4. 11. 2022

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 28. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 30. 03. 2023

Čestné prohlášení

Prohlašuji, že svou diplomovou práci „Návrh a implementace notifikační aplikace pro OS MindSphere pro společnost VDT Technology a. s.“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30. 3. 2023 _____

Poděkování

Rád bych touto cestou poděkoval Ing. Daně Vynikarové, Ph.D. za odborné vedení práce, podnětné rady a za čas, který mi v průběhu zpracování práce věnovala.

Návrh a implementace notifikační aplikace pro OS MindSphere pro společnost VDT Technology a. s.

Abstrakt

Diplomová práce se zabývá analýzou, návrhem a následnou implementací notifikační aplikace pro společnost VDT Technology a. s. Tato webová aplikace je vytvořená pomocí low-codového nástroje zvaného Mendix a implementována v OS MindSphere. Teoretická část se zabývá seznámením s konkrétními technologiemi, analýzou existujících řešení a následně je v praktické části vyvíjeno vlastní řešení notifikační aplikace na základě požadavků společnosti. Výsledná aplikace umožní uživatelům vytvářet vlastní vzhled notifikačních e-mailů za využití MindSphere API, jazyka HTML a CSS stylů na základě vytvořených pravidel v OS MindSphere.

Klíčová slova: notifikační aplikace, softwarový návrh, low-code, Mendix, MindSphere

Design and implementation of a notification application for OS MindSphere for the company VDT Technology a. s.

Abstract

The diploma thesis deals with the analysis, design and subsequent implementation of a notification application for the company VDT Technology a. s. This web application is created using a low-code tool called Mendix and implemented in OS MindSphere. The theoretical part deals with familiarization with specific technologies, analysis of existing solutions, and subsequently, in the practical part, an own notification application solution is developed based on the company's requirements. The resulting application will allow users to create their own appearance of notification e-mails using the MindSphere API, HTML language and CSS styles based on the rules created in the MindSphere OS.

Keywords: notification application, software design, low-code, Mendix, MindSphere

Obsah

1 Úvod

V dnešní internetové době se všechno velmi rychle vyvíjí a firmy potřebují udržet krok s dobou. Jejich konkurenční výhody jsou založené na nových technologiích, digitalizaci a umění optimalizace. Proto se firmy neustále snaží o optimalizaci výrobních procesů, či nákladů. K tomu jim v dnešní době může dopomoct analýza dat, digitální dvojčata či prediktivní algoritmy. Využití cloudových služeb jako je například OS MindSphere je prvotním klíčem k optimalizaci procesů, nákladů nebo například energií. Díky možnosti sběru provozních dat a jejich následné predikci je například možné včas varovat výrobce před možnou havárií jeho stroje, a díky tomu tak předejít prostojům ve výrobě, a tedy i dalším nákladům.

Včasná informovanost o vychýlení strojů z normy je nezbytná i pro optimalizaci nákladů na údržbu. Nejen však včasná informovanost, ale i srozumitelný a přesný obsah informace, který je potřeba předat je klíčový. Z tohoto důvodu byla vytvořena notifikační aplikace, ve které si uživatelé mohou vytvářet vlastní vzhledy a obsahy notifikací tak, aby reflektovaly jejich požadavky na správnou a včasnou informovanost.

První část diplomové práce se zabývá vysvětlením jednotlivých technologií, které jsou potřebné pro vyvinutí notifikační aplikace pro OS MindSphere. Druhá část diplomové práce se zabývá samotným vývojem aplikace, a to od stanovení cílů aplikace až po testování a nasazení.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem teoretické části diplomové práce je popsat základy vývoje aplikací a použité technologie pro vývoj notifikační aplikace formou literární rešerše. Dále provést analýzu a zhodnocení existujících notifikačních aplikací v OS MindSphere.

Cílem praktické části diplomové práce je provést analýzu uživatelských požadavků na notifikační aplikaci a následně vytvořit návrh vlastní notifikační aplikace, který bude implementován a otestován v OS MindSphere.

2.2 Metodika

Na základě studia odborných zdrojů budou popsány použité technologie pro vývoj aplikace formou literární rešerše. Bude provedena analýza uživatelských požadavků na notifikační aplikaci a popsána vhodnými nástroji (Use Case). Dále bude provedena analýza a zhodnocení existujících aplikací v OS MindSphere a následně bude v souladu s doporučenými postupy softwarového inženýrství vytvořen návrh vlastní notifikační aplikace (konceptuální model). Návrh aplikace bude implementován v OS MindSphere a řádně otestován.

3 Teoretická východiska

V teoretické části diplomové práce jsou nejprve popsány jednotlivé technologie, které jsou potřebné pro vyvinutí notifikační aplikace pro OS MindSphere. Následně je popsán životní cyklus aplikací a jednotlivé metodiky vývoje softwaru.

3.1 Cloud computing

Pojem cloud computing lze zjednodušeně vysvětlit jako ukládání dat, provoz aplikací a přístup k nim prostřednictvím internetu. V praxi to znamená, že jsou data a aplikace zpřístupněna odkudkoliv a jsou nezávislá na zařízení, které k nim přistupuje. Cloud computing umožňuje nejen vzdálené sdílení souborů, ale také užívání více programů několika lidmi najednou. Každý poskytovatel cloudových služeb je zodpovědný za zajištění datových center poskytující zabezpečení, paměťovou kapacitu a výpočetní výkonnost. [1]

Cloud se vyznačuje několika základními charakteristikami:

- **Služba na vyžádání**

  Cloud funguje v distribuovaném prostředí. Sdílí zdroje mezi uživatele a má schopnost splnit požadavky uživatele bez časové prodlevy.

- **Vysoká dostupnost a spolehlivost**

  Dostupnost a spolehlivost serverů je velmi vysoká, jelikož pravděpodobnost selhání infrastruktury je minimální.

- **Vysoká škálovatelnost**

  Cloud nabízí poskytování zdrojů „na vyžádání“ ve velkém měřítku, aniž by byla nutná kontrola pro maximálním zatížení.

- **Vícenásobné sdílení**

  Díky cloud computingu je možné, aby více uživatelů a aplikací pracovalo efektivněji se snížením nákladů díky tomu, že společné sdílí infrastrukturu.

- **Nezávislost zařízení na umístění**

  Cloud computing umožňuje uživatelům přístup k systémům pomocí webového prohlížeče bez ohledu na to, kde se nacházejí nebo jaké zařízení používají. Vzhledem k tomu, že je infrastruktura přístupná přes internet, uživatelé se mohou připojit odkudkoliv.

- **Údržba**

  Údržba aplikací cloud computingu je mnohem snazší, jelikož se aplikace nemusí instalovat na počítač každého uživatele, ale lze k nim přistupovat z různých míst. I tento fakt má za příčinu snižování nákladů.

- **Nízká cena**

  Použitím cloud computingu se zmenší náklady, protože pro využívání služeb cloud computingu nemusí IT společnost nastavovat vlastní infrastrukturu a platit podle využití zdrojů. [2]

Obrázek č. 1 – Schéma Cloud Computingu [3]

3.1.1 Historie cloud computingu

Historie cloud computingu sahá až do 60. let minulého století. V roce 1961 jako první svou myšlenku prezentoval John McCarthy, profesor z prestižní univerzity MIT. Jeho představa byla taková, že jednoho dne bude možné prodávat výpočetní sílu jako běžnou veličinu, stejně či podobně jako třeba elektřinu nebo plyn.

Pojem „Cloud Computing“ se objevil v roce 1997 v jedné z přednášek Ramnatha Chellapa. Pojem „cloud“ je pouze popisným vyjádřením schematického obrázku, ve kterém je nakreslena infrastruktura poskytovatele. Historicky je v telekomunikacích používán pojem pro zobrazení telekomunikační sítě právě pojem cloud. IT si toto zobrazení vypůjčilo, z toho důvodu, že se v telekomunikacích koncové stanice připojené do internetu zobrazují jako krabičky připojené do oblaku s nápisem Internet. [4]

3.1.2 Hlediska cloud computingu

Existují dvě základní hlediska, kterými lze porovnávat jak výhody, tak nevýhody cloud computingu.

3.1.2.1 Ekonomické hledisko

Pokud má firma vlastní infrastrukturu, velmi často se objevují skryté výdaje. Jelikož je občas nutné přeinstalovat server po nečekaném výpadku či dokoupit disk do diskového pole. U cloud computingu jsou tyto skryté náklady zcela eliminovány. Služby cloud computingu jsou placené a je na provozovateli, aby je poskytl.

S cloud computingem je možné dosáhnout úplné nezávislosti na vlastních IT zaměstnancích. Předchází se situacím, kdy je společnost závislá na jednom IT zaměstnanci, jelikož právě tento zaměstnanec jako jediný ví, jak je nastaven například Exchange server, zná hesla a je z nějakého důvodu nedostupný v potřebných chvílích. Obzvláště u menších firem je výhoda nezávislosti na IT zaměstnancích opodstatněnější, když firma nemá vlastní IT, ale najímá si externí pracovníky.

Za velkou přednost cloud computingu lze považovat možnost dynamicky měnit kapacitu služeb, která je hlavně u větších poskytovatelů takřka neomezená. Například po vyhrání velké zakázky firma potřebuje rozšířit svou kapacitu o 100 nových poštovních schránek. S cloud computingem je to možné do několika minut. S vlastní infrastrukturou to není až tak jednoduché a pravděpodobně bude nutné dokoupit další server. Nebo naopak po propuštění 50 zaměstnanců je možné dynamicky upravit požadovaný počet poštovních schránek a je ihned možné dosáhnout ušetřených výdajů. S vlastní infrastrukturou je infrastruktura stále přizpůsobená původní kapacitě, která je v danou chvíli zbytečná. Za zvyšování kapacity si poskytovatelé neúčtují žádné další poplatky, ale to ovšem neplatí ve všech případech i při snižování kapacity, kdy si někdy poskytovatelé účtují penále. Velké ekonomické výhody mohou také zaznamenat sezónní firmy. Například e-shop si může pronajmout větší výpočetní výkon u serverů využívaných ve formě cloud computingu (typicky hostingu) jen na pár měsíců a po zbytek roku si vystačí s menší kapacitou.

Z dlouhodobého hlediska cena cloud computingu může přesahovat náklady, které jsou vynaloženy na realizaci svépomocí. Týká se to například pronájmu infrastruktury, kde existuje určitá přirážka za to, že poskytovatel cloud computingu v SLA garantuje

obvykle poměrně vysokou (nejčastěji 99,9 %) dostupnost, za jejíž nedodržení je sankcionován (obvykle slevou na měsíčním/ročním paušálu). [4]

3.1.2.2 Technické hledisko

Největší předností cloud computingu je jeho rychlost nasazení, ať už se jedná o některé aplikace či infrastrukturní služby. Nasazení je možné dosáhnout prakticky ihned. Stačí jen několik málo kroků, jakými jsou například vybrání poskytovatele, vyplnění registračního formuláře, případně přímo rovnou danou službu zaplatit a službu je možné ihned používat. Pokud by firma chtěla nasadit stejnou infrastrukturu ale vlastní, zabere to mnohem více času, a to v rámci týdnů či dokonce měsíců. Jelikož je potřeba vybrat dodavatele hardwaru, softwarovou platformu, vyplnit objednávku a následně čekat několik týdnů na dodání potřebného hardwaru. Následně je také potřeba vše nainstalovat, nakonfigurovat, otestovat a až teprve tehdy je možné začít pracovat.

Další výhodou cloud computingu je to, že odpadá veškerá péče o serverovou část infrastruktury. Není potřeba řešit žádné výpadky, záplatování a ani dokonce není potřeba řešit hackerské útoky. Jediné, co je v některých případech potřeba řešit je podpora vlastních zákazníků, pokud poskytovatel tuto podporu nenabízí sám. Pokud přijde nová verze softwaru, není potřeba nic zálohovat, instalovat, migrovat, kontrolovat anebo dokonce kupovat. Vše je tedy neustále aktualizované a díky tomu je možné přistupovat vždy k nejnovějším funkcím. O aktualizace se stará poskytovatel.

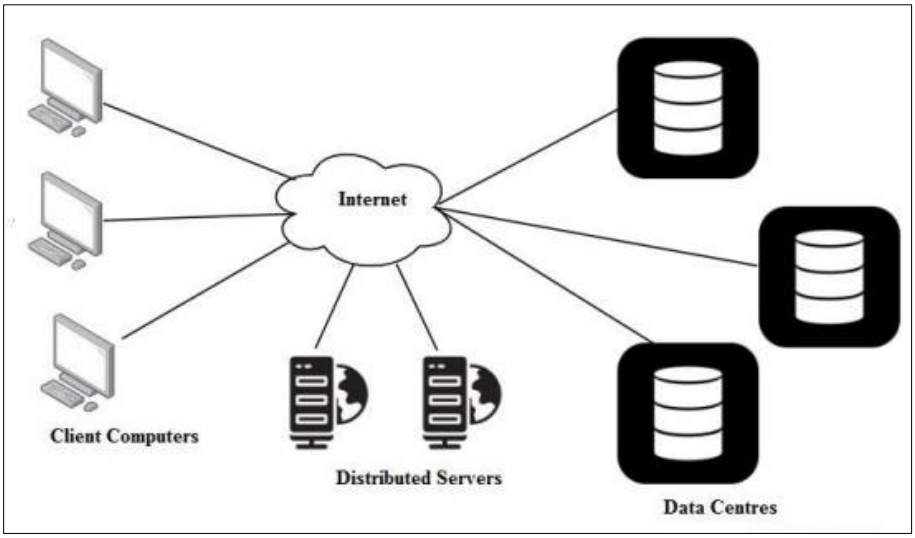
Z technického hlediska je také velkou výhodou Service Level Agreement neboli smluvně garantovaná dostupnost služby. Neexistuje mnoho firem, které by měly vlastní systémy nastavené tak, že by IT oddělení garantovalo dostupnost některých služeb. Pokud chce tedy firma dosáhnou vysoké míry dostupnosti, je potřeba vyhledat poskytovatele, který je schopný nejen SLA nabídnout, ale také i dodržet.

Firma využívající služby cloud computingu se může obávat toho, že k jejich datům má fyzicky přístup daný poskytovatel, tedy někdo cizí. Je možné, že správce daného serveru má přístup k souborům, e-mailům a podobně. Avšak nikdo jiný kromě samotného poskytovatele tento přístup nemá. Je nutné podotknout, že vše je šifrované a zabezpečené, a to pravděpodobně lépe než v případě vlastních serverů. I když jsou všechna data zašifrována, může se stát, že při sofistikovanějším útoku se k nim někdo dostane. To samé

je ale možné i v opačném případě, kdy jsou data na počítačích nebo lokální síti připojené k internetu. Jen se pravděpodobně bude jednat o jiný způsob provedení útoku, díky kterému může daný útočník získat data. [4]

3.1.3 Komponenty Cloud Computingu

Existují tři hlavní komponenty Cloud Computingu. Těmi jsou klienti, datacentra a distribuované servery. [5]



Obrázek č. 2 – Komponenty Cloudu [5]

3.1.3.1 Klienti

Klienti jsou v Cloud Computingu zařízení, díky kterým se koncový uživatel připojí ke svým datům uložených na cloudu. Lze je rozdělit do třech kategorií na tenký klient, tlustý klient a chytrý klient. Jsou jimi například chytré telefony, stolní počítače nebo notebooky. Prakticky se může jednat o jakékoliv zařízení, které je schopné připojení k internetu. [5]

Tenký klient je téměř jako klasický stolní počítač. Hlavním rozdílem je ten, že nemá pevný disk a také to, že výkon tenkého klienta by pro plnohodnotný běh operačního systému nestačil. Proto běh operačního systému zajišťuje centrální server. Jedná se o velmi výkonný počítač s dostatkem úložného prostoru, na kterém pracuje více uživatelů a přistupují k němu právě skrze tenké klienty, kteří mají pouze za úkol zobrazení dění na obrazovce právě daného vzdáleného počítače. [6]

Tlustý klient je také nazýván jako těžký klient. Jedná se o plně vybavený počítač, který je připojen k síti. Na rozdíl od tenkého klienta, tlustému klientovi nechybí pevný disk a další funkce, takže je schopný fungovat i bez ohledu na to, zda je připojen k síti či nikoliv.

Tlustý klient je klientem pouze tehdy, když je na síti a je připojen k serveru. Ten má možnost poskytnout tlustému klientovi programy a soubory, které nejsou uloženy na pevném disku místního počítače. [7]

Chytrý klient je kombinací výhod tenkého a tlustého klienta. Chytrý klient obsahuje určitou logiku a drží data. Díky tomu může pracovat offline a ve chvíli, kdy se naváže spojení, tak daná data synchronizuje. Využívá místní systémové zdroje jako je např. paměť, procesor a diskový prostor, ale může využívat a komunikovat i s připojenými zařízeními. [8]

3.1.3.2 Datacentra

Datová centra jsou místa, ve kterých je zabudovaná síť výpočetních a úložných serverů a dalších IT technologií. Tato místa jsou budována pro nepřetržitý provoz a jejich hlavní cíl je, aby servery a jiná technologická zařízení měly přístup ke stabilnímu a bezproblémovému provozu, a to zcela nezávisle na vnějších vlivech. Zjednodušeně je možné říct, že jde o uchování a zabezpečení klíčových dat na jednom místě. [9]

3.1.3.3 Distribuované servery

Distribuované servery mají stejnou funkci jako datová centra s tím rozdílem, že jsou servery provozovány na různých místech. Jedná se o přínos pro poskytovatele cloudových služeb, jelikož díky možnosti propojení mnoha serverů na širším území se pružnost a bezpečnost poskytovaných služeb mnohonásobně zvýší. Pokud by například chtěl poskytovatel rozšířit kapacitu, má možnost přidat nový server jako součást cloudu, a to bez ohledu na umístění daného serveru. Koncový uživatel nepozná, kde jsou dané servery umístěny. Zda jsou vedle sebe nebo jsou vzdálené tisíce kilometrů. [10]

3.1.4 Modely nasazení

Modely nasazení jsou kategorizovány do čtyř základních typů cloudu. Jedná se o veřejný cloud, privátní cloud, hybridní cloud a komunitní cloud. Všechny typy nasazení fungují na principu cloudových serverů, které spravují softwarové aplikace. [11]

3.1.4.1 Veřejný

Aktivity veřejného cloudu jsou spravovány externími stranami poskytováním odpovídajících cloudových služeb veřejnosti prostřednictvím internetu, a tyto služby jsou přístupné prostřednictvím určitých typů plateb. Díky rozsáhlým funkcím tyto služby snižují

cenu IT infrastruktury a lépe fungují pro správu obrovského množství dat na samotné místní infrastruktuře.

Výhody veřejného cloudu:

- Poskytuje škálovatelnost
- Implementace a údržba jsou nákladově efektivní
- Zvýšená spolehlivost, která předchází k přerušení služeb
- Veřejná cloudová platforma snadno hostuje služby SaaS, IaaS a PaaS, které umožňují snadnou dostupnost těchto služeb přes internet.
- Platforma je zcela nezávislá na umístění [11]

3.1.4.2 Privátní

Privátní cloudy jsou ve větším případě implementovány ve velkých firmách a to tak, aby vyvíjely a spravovaly své informační centra pro vlastní požadavky a provoz. Jinými slovy privátní cloudy jsou vytvořeny pro výhradní použití jednoho klienta a poskytují plnou kontrolu nad daty, zabezpečením a kvalitou služeb. Soukromé cloudy lze budovat a spravovat vlastní IT organizací nebo poskytovatelem cloudu. Výhodou je zvýšená bezpečnost a plná kontrola nad daty. Nevýhodou je však cena, omezenost kapacity a nutnost IT podpory. [12]

3.1.4.3 Hybridní

Jedná se o integraci veřejného i soukromého cloudu a představuje tak střední cestu mezi danými modely. Hybridní cloud umožňuje ukládat do veřejného cloudu méně citlivé informace a všechny klíčové informace do cloudu privátního. Jedná se o typ cloudu, který je dražší nežli cloudy veřejné, ale zároveň nabízí vlastnosti cloudů privátních s nižší cenou. [11]

3.1.4.4 Komunitní

Jedná se o velmi zvláštní případ veřejného cloudu. Tento typ cloudu je poskytován specifické komunitě podniků, které mají společné zájmy. Vzniká v momentě dohody více subjektů za podmínek, které si mezi sebou dohodnou a ty se následně musí respektovat. Výhodou tohoto modelu je cena, avšak pouze v případě porovnání s privátním cloudem. Naopak při porovnání s veřejným cloudem je výhoda bezpečnost. [13]

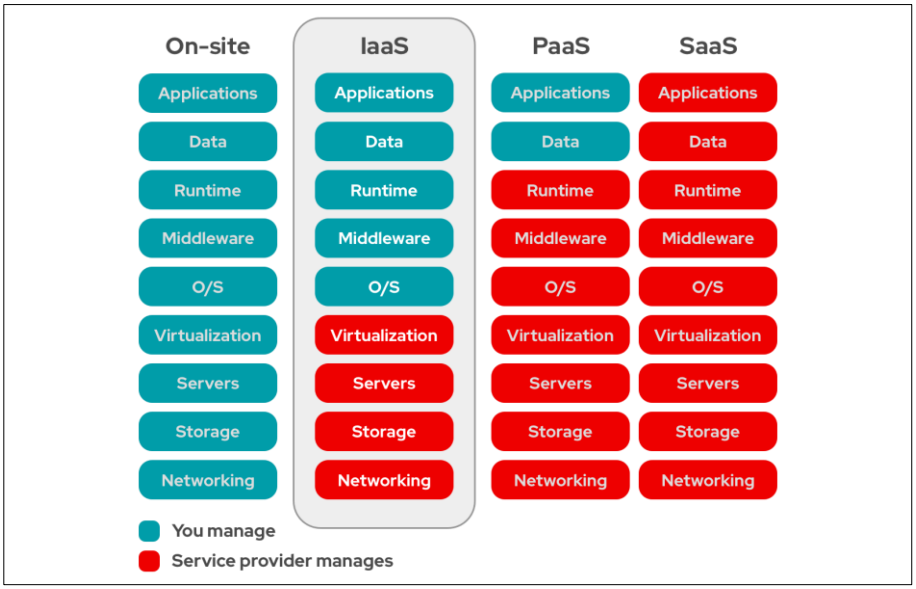
3.1.5 Distribuční modely

Výběr správného distribučního modelu je rozhodujícím faktorem úspěchu pro poskytování cloudových řešení. Aby bylo možné vybrat správný model nebo kombinaci modelů, je potřeba plně porozumět tomu, co každý z modelů nabízí, jaké odpovědnosti přebírají poskytovatelé cloudových služeb a jaké odpovědnosti přebírá spotřebitel cloudových služeb.

Každý model cloudové služby poskytuje úroveň abstrakce, která snižuje úsilí, které spotřebitel služby vyžaduje při budování a nasazení systémů. V tradičním místním datovém centru musí IT tým vše vybudovat a spravovat. Ať už tým vytváří řešení od nuly nebo kupuje komerční softwarové produkty, musí instalovat a spravovat servery, vyvíjet a instalovat software a také zajistit, aby byla aplikována správná úroveň zabezpečení. Dále je také potřeba rutinně aplikovat záplaty a mnoho dalšího. Každý model cloudových služeb poskytuje spotřebitelům cloudových služeb větší agilitu, aby se mohli více věnovat svým obchodním problémům a méně času infrastruktuře. [14]

3.1.5.1 IaaS

„Infrastructure as a Service“ neboli infrastruktura jako služba. Jedná se o nejzákladnější typ služby, ve kterém je poskytována zákazníkovi infrastruktura. Tím jsou myšleny zejména procesory, servery (fyzické i virtuální) a prostředky uložiště. V posledních letech se stává IaaS čím dál více populární, a to především u začínajících podniků a rychle rostoucích divizí velkých společností, které se snaží vyvinout své vlastní aplikace, ale vyhýbají se investicím a údržbě, kterou infrastruktura vyžaduje. [15]

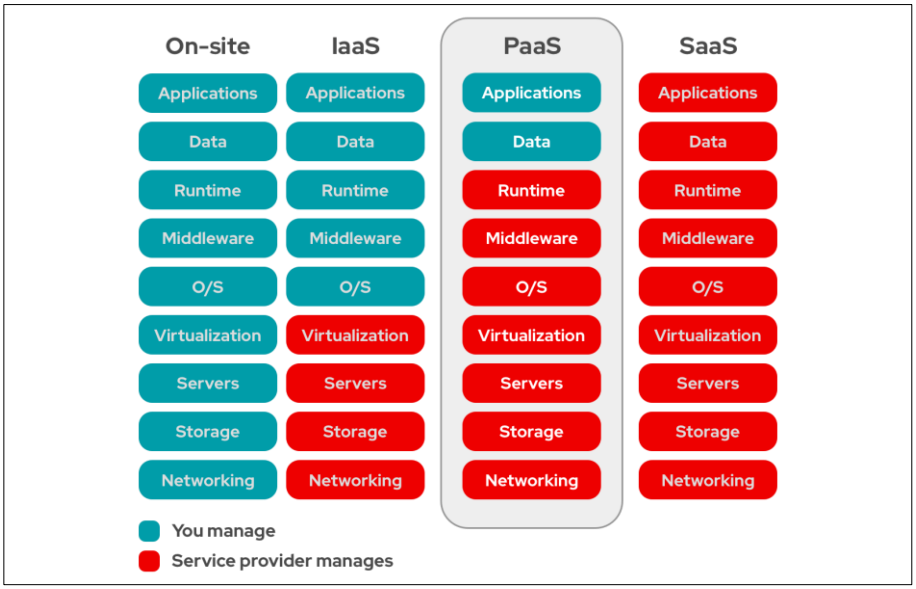


Obrázek č. 3 – Distribuční model IaaS [16]

Výhodou tohoto modelu je, že je velmi flexibilní, snadno škálovatelný a cenově dostupný. Za nevýhodu lze považovat bezpečnost. Mezi největší poskytovatele tohoto modelu se řadí například Amazon Web Services, Microsoft nebo Alibaba Cloud. [15]

3.1.5.2 PaaS

„Platform as a Service“ znamená platforma jako služba. Jedná se o model, který podle ročních příjmů dosahuje nejmenších zisků. Mnoho odborníků a analytiků však očekává, že se PaaS stane jedním z nejdůležitějších a nejrychleji rostoucích segmentů v rámci podnikových technologií. PaaS obecně zahrnuje základní úložiště, sítě a virtuální servery, které jsou k dispozici na základě průběžných plateb, ale také poskytuje mnoho nástrojů a dalšího softwaru, který vývojáři potřebují při vytváření aplikací. To zahrnuje middleware, správu databází, operační systémy a vývojové nástroje specifické pro aplikace. Jinými slovy, PaaS nabízí stejnou jednoduchost a flexibilitu jako IaaS, ale rozšiřuje podporu na celou vývojářskou komunitu. Za výhodu lze považovat to, že se uživatel nemusí starat o instalace nebo aktualizace aplikací a může se zaměřit pouze na vývoj. [15]



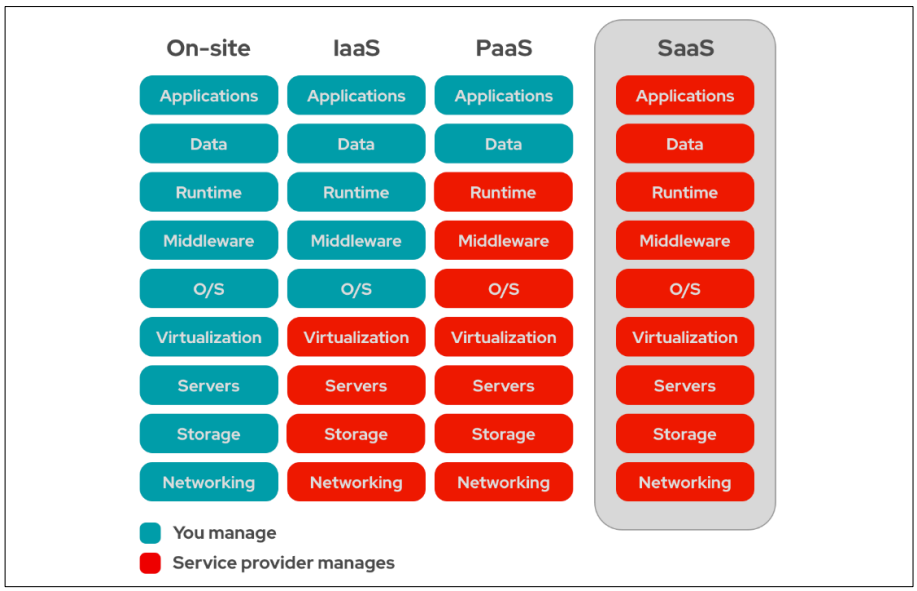
Obrázek č. 4 – Distribuční model PaaS [17]

3.1.5.3 SaaS

„Software as a Service“ se dá přeložit jako software jako služba. Jedná se o jednoznačně nejrozšířenější a nejdominantnější ze všech modelů cloud computingu a v blízké budoucnosti tomu tak i pravděpodobně zůstane. Ve své podstatě model SaaS dodává aplikace přímo z cloudu jednotlivým uživatelům, kteří k aplikacím přistupují prostřednictvím webových prohlížečů jako je Chrome, FireFox nebo Safari. Poskytovatelé SaaS nabízejí doslova kompletní službu. Organizace všech velikostí tak již nemusí investovat do drahých hardwarů či operačních systémů. Dále také nemusejí platit žádné IT pracovníky za údržbu infrastruktury nebo za odstraňování problémů s aplikacemi. To vše poskytuje dodavatel SaaS.

Avšak ne všechny typy aplikací jsou vhodné pro využití tohoto modelu. Organizace s velmi specifickým zaměřením, jako jsou například výrobci jaderných reaktorů, budou spíše preferovat vlastní aplikace, které budou více přizpůsobené jejich potřebám. Pro organizace, které využívají více zobecněné aplikace je SaaS velmi často nejlepším řešením.

Nadšení pro model SaaS se odráží i na výsledcích. Za rok 2018 byly dvě třetiny financí utracené za cloudové služby právě za model SaaS. A předpokládá se, že do šesti let dosáhne trh s řešením SaaS 186 miliard dolarů s ročním růstem 22 %. [15]



Obrázek č. 5 – Distribuční model SaaS [18]

3.1.5.4 XaaS

„Anything as a service“ neboli cokoliv jako služba je souhrnný termín, který označuje dodání čehokoli jako službu. Zahrnuje mnoho produktů, nástrojů a technologií, které dodavatelé dodávají uživatelům jako službu přes síť. Tento zastřešující termín se vztahuje k nabídkám služeb, které jsou dostupné podle potřeby a jsou financovány pomocí cenového modelu cloud computingu s průběžnými platbami. Nabídky XaaS lze podle potřeby škálovat pomocí IT služeb poskytovaných na vyžádání poskytovatelem spravovaných služeb. Kromě již zmiňovaných třech modelů IaaS, PaaS a SaaS, existuje mnoho dalších příkladů XaaS, jako jsou například:

- „Authentication as a service“ zkráceně AaaS, který využívá cloudové služby pro správu identit a přístupu
- „Containers as a sevice“ zkráceně CaaS, který umožňuje nasazení a správu kontejnerů pomocí virtualizace založené na kontejnerech
- „Device as a service“ zkráceně DaaS je typ služby, kdy dodavatel třetí strany nabízí počítače, chytré telefony a další mobilní výpočetní zařízení jako placenou službu
- „Function as a service“ zkráceně FaaS umožňuje cloudovým zákazníkům vyvíjet aplikace, nasazovat funkce a účtovat poplatky pouze tehdy, když se funkce spustí

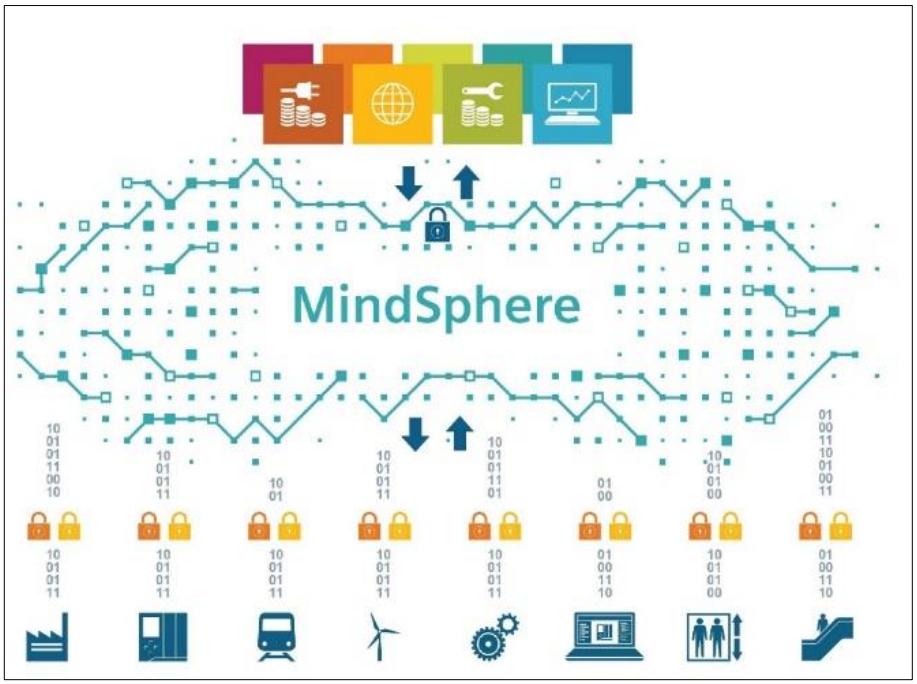
- „Database as a service“ neboli DBaaS, který poskytuje přístup k databázovým platformám prostřednictvím cloudu. Poskytovatelé veřejného cloudu jako AWS a Azure tento typ nabízejí
- „Storage as a service“ neboli STaaS, poskytuje aplikační, datové a zálohovací úložné systémy v cloudu [19]

3.2 MindSphere

MindSphere je otevřený operační systém a cloudové řešení pro průmyslový internet věcí (IoT), který byl vyvinut společností Siemens. V roce 2016 byl představen jako digitalizační platforma pro všechna odvětví. MindSphere propojuje všechny přístroje, stroje nebo senzory. Může se jednat například o vybavení chytrých továren nebo zařízení, která slouží k řízení dopravy a dopravní infrastruktury. Také to mohou být technologické celky v chemii či v energetice, nebo „pouze“ připojený senzor či počítač, který sbírá data. Všechna tato zařízení mají společné to, že vytvářejí velké množství dat (big data). S těmito daty umožňuje MindSphere dále pracovat a za využití aplikací a pokročilých analytických funkcí je proměnit v chytrá data. [20]

3.2.1 Architektura

MindSphere je založena na třech základních vrstvách.



Obrázek č. 6 – Vrstvy MindSphere [21]

3.2.1.1 MindConnect

Spodní vrstva systému je zaměřená k připojování zařízení, ze kterých se sbírají data. Pro tuto úroveň existuje mnoho nástrojů. Jedná se o nástroje softwarové ale i hardwarové.

Pomocí těchto nástrojů lze jednoduše připojit systémy, stroje nebo zařízení, a to jak od společnosti Siemens, tak i třetích stran, a to mnohdy velmi snadným způsobem plug & play. [22]

3.2.1.2 MindSphere

Střední vrstva systému je složena ze samotného cloudového úložiště. V této oblasti společnost Siemens spolupracuje s firmami, jako je SAP nebo Amazon Web Services, které poskytují vlastní cloudové služby. Doposud je MindSphere provozován pouze na cloudech veřejných, avšak připravuje se i řešení pro soukromé cloudy. [22]

3.2.1.3 MindApps

Vrstva aplikační je v tomto systému nejvyšší vrstvou. Na budování této vrstvy se podílí jak společnost Siemens, tak i třetí strany. Hotové aplikace jsou zpřístupněny zákazníkům přes internetový obchod MindSphere Store s kompletní nabídkou aplikací od různých vývojářů. Umožněno je také vyvinout vlastní aplikace bez nutnosti publikování aplikací na MindSphere Store. [22]

3.2.2 Rozdělení

Platforma MindSphere se rozděluje na tři MindAccess plány podle jejich specifických využití. [23]

3.2.2.1 Developer Plan

V tomto prostředí se vyvíjejí aplikace, které se připravují do produkce. Samotné prostředí není pro produkci určeno. Aplikace se vyvíjejí pomocí speciálních nástrojů, které je možné zakoupit v MindSphere Storu. Další možností je vývoj aplikací pomocí low-code platformy Mendix. Umožněno je však i psát aplikace s využitím API, díky kterým lze přistupovat k jednotlivým datům, uložených v MindSphere. [23]

3.2.2.2 Operator Plan

Po vyvinutí aplikace je možné z developer plánu přesunout aplikaci do operator plánu, kde se dále monitoruje a spravuje. Z tohoto prostředí je možné vlastní aplikace přesunout do Value plánu, nebo je distribuovat do MindSphere Store. [23]

3.2.2.3 IoT Value Plan (User Plan)

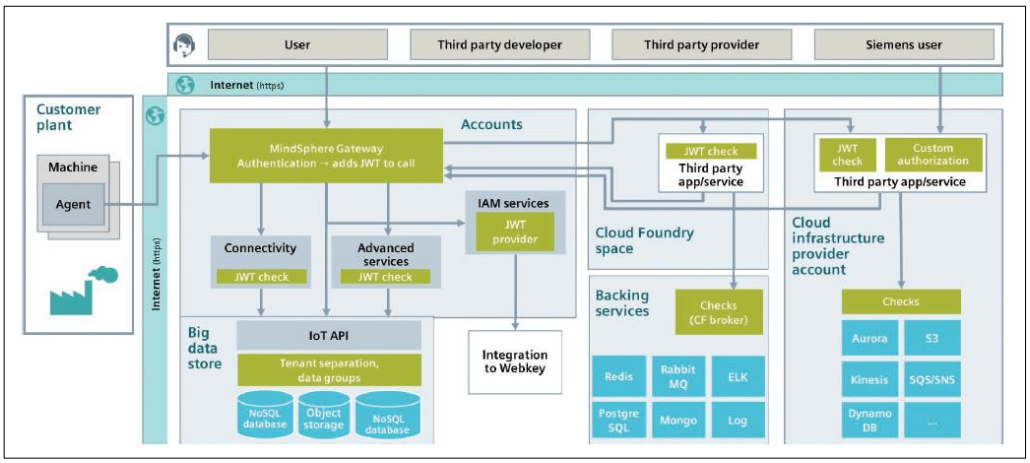
Toto prostředí je zaměřené primárně na sběr dat z jednotlivých zařízení, strojů a systémů konkrétního zákazníka. Zákazníkovi je umožněno si zakoupit jakoukoliv aplikaci z MindSphere Store a využívat tak její analytické, prediktivní a další funkce. [23]

3.2.3 Zabezpečení

Společnost Siemens řadí špičkovou bezpečnost mezi nejvyšší priority. Specializovaný bezpečnostní tým MindSphere má za úkol ochranu dat po celou dobu jejich životního cyklu, a to od připojení zařízení, sběru dat, uchování údajů až po vyřazení z provozu. [24]

3.2.4 Bezpečnostní architektura

Silné řešení IoT vyžaduje podrobné zabezpečení architektury systému, které může efektivně představovat vícevrstvé zabezpečení v rámci řešení. Platforma MindSphere je navržena tak, aby to byla efektivní, odolná a škálovatelná infrastruktura. Funkce jako jsou přísná správa přístupu, separace prostředí, zabezpečení sítě, filtrované komunikační kanály prostřednictvím řídicí brány MindSphere Gateway jsou pro architekturu MindSphere zásadní. Bezpečnostní funkce MindSphere jsou nejen zřejmé, ale také jsou rozhodující pro fungování celého systému. Obrázek viz níže, znázorňuje bezpečnostní prvky architektury MindSphere. [24]



Obrázek č. 7 – Bezpečnostní architektura MindSphere [25]

3.2.5 Správa identit a kontrola přístupu

Model práv a přístupová politika MindSphere je spojení těch nejmodernějších, které jsou založené na pravidlech a rolích. Tento model odpovídá doporučeným normám Národního institutu standardů a technologie (NIST). Model domény pro správu identity a přístupu určuje, jak zákazník může autorizovat uživatele a správce v MindSphere.

Citlivá data proti zveřejnění chrání především zachování jasného postupu pro autorizaci a také důsledné ověřování uživatele s příslušnými oprávněními. Správa přístupu mezi účty je pokaždé spuštěna přes JSON Web Token (JWT), jak je možné vidět na obrázku číslo 7. [26]

3.2.6 Přístupová ověření a autorizace

V centru pozornosti je poskytnutí vhodného přístupu k údajům správa identity a kontrola přístupu (IAM – identity and access management). MindSphere podporuje multifaktorové ověřování (MFA – *Multi-factor authentication*) k potvrzení totožnosti uživatelů, kteří se pokoušejí o přístup do systému. Hesla použitá v MFA se řídí mezinárodními průmyslovými standardy pro požadovanou sílu. Ověření a autorizace na úrovni rozhraní programování webových aplikací (API – *Application* Programming Interface) a webových aplikací závisí na informacích o autentizaci a autorizaci poskytovaných komponentami MindSphere IAM. [26]

3.2.7 Zabezpečení komunikace

Veškerá komunikace, která probíhá mezi MindSphere a klientem je skrze veřejné koncové body, které jsou zabezpečeny prostřednictvím TLS v.1.2, podle osvědčených postupů v odvětví komunikace. Používají se spolehlivé certifikáty x509 od Siemens Trust Center, které jsou prověřeny Evropským institutem pro telekomunikační standardy (ETSI). MindSphere používá nejmodernější šifrování s perfektní dopřednou bezpečností (PFS - Perfect Forward Secrecy), aby se zabránilo opakovaným útokům. [26]

3.2.8 Připojení

Pro možnosti připojení se využívají nástroje MindConnect, které nabízí širokou škálu možností pro propojení strojů, systémů, technologických zařízení a produktů od různých výrobců. K dispozici jsou jak hardwarové, tak i softwarové nástroje. [27]

3.2.8.1 Možnosti připojení

Možnosti připojení do MindSphere:

- MindConnect Nano
- MindConnect IoT 2040
- MindConnect Lib/API
- MindConnect IoT Extension
- Vestavěné nástroje pro integraci řídicích systémů Simatic S7 nebo Sinumerik
- Integrace do podnikových systémů CRM, ERP, MES, apod.
- A další možnosti se již připravují [27]

3.2.8.2 MindConnect LIB

MindConnect Lib je softwarová knihovna, která má možnost připojit zařízení od společnosti Siemens, ale i zařízení třetích stran s MindSphere. Jelikož je řešení pro každé odvětví i každou společnost různorodé, poskytuje MindConnect LIB sadu pro vývoj softwaru, díky které je možné naprogramovat konektivitu, specifickou pro zákazníka. Této konektivity je možné dosáhnout skrze MindSphere API, díky kterému je možné propojit zařízení s MindSphere, a to bezpečně a efektivně. Bezpečné přenášení dat je zajištěno pomocí MindSphere API a jedinečných bezpečnostních tokenů ověřující autentizaci. Šifrovaná komunikace s MindSphere je prostřednictvím zabezpečení transportní vrstvy TLS 1.2 s šifrovacím klíčem délky 128 až 256 bitů. [28]

3.2.8.3 MindConnect Nano, IoT 2040

MindConnect Nano i IoT 2040 jsou plug & play zařízení, které dokážou velmi snadno a spolehlivě načíst data z připojeného systému, předběžně si je zpracovat a připravit pro šifrovaný přenos do MindSphere. Zašifrovaná data se pak bezpečně přenesou do platformy MindSphere, kde je možné provést další analýzu. Pracují na operačním systému Linux, který je speciálně upravený pro konektivitu právě s MindSphere. Umožňují pouze jednosměrnou komunikaci a podporují různé protokoly jako například protokol S7 nebo OPC UA. [27]

3.2.9 Aplikace

Veškeré dosud vytvořené dostupné aplikace jsou k dostání v MindSphere Store. Mnoho dalších aplikací je aktuálně ve vývoji, a to nejen společností Siemens. Vyvíjet

a nabízet zákazníkům své vlastní aplikace pro MindSphere mohou i třetí strany. Také i zákazníci si mohou vyvíjet své vlastní aplikace. Díky otevřenému prostředí pro vývoj aplikací je potenciál MindSphere téměř neomezený. [27]

3.2.10 Využití MindSphere

Do světa digitálních služeb otevírá bránu právě platforma MindSphere. Služby, které svým zákazníkům poskytuje, jim přináší značnou konkurenční výhodu. Nejen že zvyšují dostupnost, ale hlavně zlepšují produktivitu a efektivitu jednotlivých strojů, systémů i výrobních závodů po celém světě. Využití má ve všech oblastech, od průmyslové výroby, přes energetiku, železniční a silniční dopravu, logistiku, až po chytrá města. [27]

Obrázek č. 8 – Oblasti využití [28]

Kromě již zmiňovaných výhod lze díky platformě MindSphere:

1. optimalizovat spotřebu energií
2. zvyšovat sledovatelnost výroby
3. minimalizovat náklady na údržbu
4. snižovat výrobní náklady
5. zvyšovat efektivitu a produktivitu
6. zvyšovat ziskovost díky:
    a. analýze dat
    b. provádění simulací
    c. detekci a následnou opravou strojů před poruchou
    d. monitorování dat v reálném čase [29]

Obrázek č. 9 – Chytrá továrna [30]

3.2.11 Ekosystém

Aby bylo možné provést úspěšnou digitální transformaci, společnost Siemens podporuje rozsáhle partnerské sítě v rámci všech odvětvích, a to jak na globální, tak i na lokální úrovni. S partnery sdílí nejen znalosti daných oborů ale hlavně rozsáhlé znalosti v IT a datové analytice. MindSphere nabízí svým partnerům jedinečnou možnost, jak se podílet na digitalizaci celého hodnotového řetězce, bez ohledu na obor jejich podnikání či velikost. Otevřenost platformy na všech úrovních je považováno za hlavní výhodu. Díky tomu je umožněno uživatelům se jednoduše a odkudkoliv připojit ke všem svým zařízením. Pomocí otevřených aplikačních rozhraní (API) mohou vyvíjet, instalovat a distribuovat vlastní MindSphere řešení nebo je sdílet v rámci MindSphere komunity. [27]

3.3 Internet věcí IoT

Intertnet of Things zkráceně IoT je technologií, která realizuje představy o světě, kde společně všechna elektronická zařízení spolu navzájem komunikují, vyměňují si data a automatizují nejen profesní, ale i osobní život. IoT lze tedy snadno vysvětlit jako ekosystém chytrých zařízení, počítačů či strojů, které jsou mezi sebou schopny vzájemně komunikovat bez zásahu lidské síly. Může se tedy jednat i o zcela obyčejnou elektroniku, jako jsou například ledničky, hodinky nebo teploměry, které pokud jsou doplněny o operační systém a připojení k internetu, dostanou zcela nové možnosti využití a benefity pro běžné činnosti. Toto doplnění lze popsat jako digitální transformaci, která z původně obyčejných (hloupých) zařízení dělá chytrá neboli smart zařízení. [31]

3.3.1 Historie

V roce 1999 definoval pojem Intertnet of Things britský inženýr Kevin Ashton, který je známý díky vytvořené technologii RFID (identifikace zboží). Jedná se o alternativu pro čárové kódy. Avšak samotná myšlenka navzájem provázaných zařízení se datuje až do roku 1832, kdy byl vynalezen telegraf. První výskyty IoT bylo možné spatřit již v 60. letech 20. století, kdy společnost Coca-cola vyvinula automat, který bez zásahu lidské síly dokázal monitorovat stav chlazení umístěných nápojů vně automatu, a v případě nedostatku nápojů hlásit potřebu doplnění zboží.

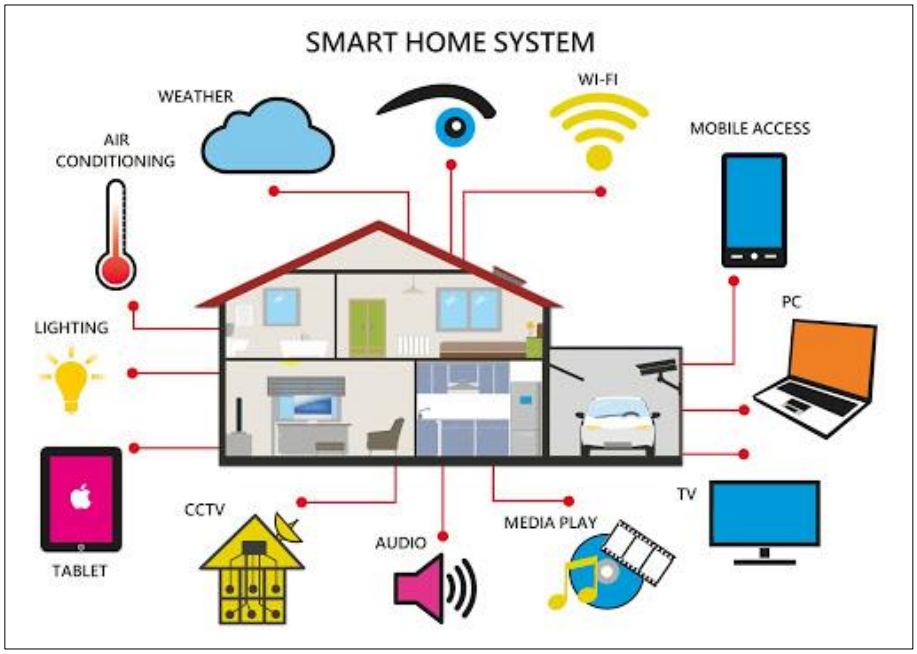
Dalším milníkem jsou považována 90. léta 20. století, kdy John Romkey prováděl experimenty s toustovačem, který za pomocí protokolu TCP/IP připojil k internetu. Ve stejné době na univerzitě Cambridge použili webkameru jako nástroj k tomu, aby mohli sledovat stav kávovaru. Díky tomu se tak zaměstnanci mohli ujistit, že jejich káva je již připravená. Mezi širokou veřejnost se termín IoT setkal až v 21. století, kdy se konala první oficiální konference na toto téma, a to v roce 2008 ve Švýcarsku.

Hlavním bodem zlomu rozšíření IoT nastal v roce 2011, kdy přišla technologie IPv6, která zajišťovala dostatek IP adres a díky tomu byl umožněn masový nárůst IoT po celém světě. [31]

3.3.2 Chytrá domácnost

V dnešní době se s termínem IoT lze nejčastěji potkat v oblasti chytré domácnosti, neboli SmartHome. Cílem chytré domácnosti je sjednotit jednotlivé IoT zařízení do jednoho centrálního místa tzv. hubu, přes který je možné všechna tato zařízení ovládat.

Skrze jedinou mobilní aplikaci je tak umožněno mít přístup ke všem připojeným zařízením v domácnosti, a to v reálném čase. [31]



Obrázek č. 10 – Chytrá domácnost [32]

3.3.3 Chytré město

Další oblastí, ve které je možné snadno narazit na termín IoT je chytré město (SmartCity). Jde o stejný koncept jako v případě chytré domácnosti akorát místo domácnosti se jedná o celé město a místo domácích spotřebičů se jedná o dynamické křižovatky, chytrou dopravu, chytré budovy a další. K těmto technologiím přispívá i fakt, že se počet chytrých zařízení neustále rozrůstá, a to velmi vysokým tempem. V roce 2015 bylo v oběhu 15 miliard chytrých zařízení. V roce 2019 bylo chytrých zařízení 26,66 miliard a předpokládá se, že do roku 2025 bude nárůst na 75,44 miliard zařízení. [31]

Obrázek č. 11 – Chytré město [33]

3.3.4 IoT a cloud

Každé IoT zařízení generuje obrovské množství dat. Data se sbírají a zpracovávají a na základě těchto zpracovaných dat je následně možné provádět analýzy, které dokážou detekovat a předcházet nežádoucím problémům například se zdravím. Naopak v oblasti průmyslu díky analýze dat lze předcházet například k poruchám strojů, které by mohly mít za následek nežádoucí odstávku. K účelům analýzy se v převážných případech používá cloud, kam jsou data sbírána a následně analyzována za využití cloudových zdrojů. Z hlediska architektury a principu fungování lze IoT technologie shrnout do několika bodů:

- Senzory a zařízení – monitoring efektivity činnosti produktu
- Konektivita – přenos dat do IoT platformy (cloud) a zároveň párování zařízení v síti skrze technologie BlueTooth, WiFi nebo WAN
- Zpracování dat – IoT zařízení generují ohromné množství dat tzv. big data
- Uživatelská rozhraní – aplikace pro koncové uživatele, které poskytují vizualizaci ať již surových, tak i analyzovaných dat [31]

3.4 API

Appliaciton Programming Interface neboli API je soubor procedur, funkcí, protokolů a knihoven, který využívají programátoři k tvorbě webových i mobilních aplikací a k tvorbě softwaru. Hlavní podstatou API je zajištění komunikace mezi dvěma platformami, které si vzájemně vyměňují data. Umožňují integrovat do vlastních webů či softwaru již naprogramovaná řešení a díky tomu je ušetřen programátorský čas a tím i peníze. V reálném světě je možné si API představit v roli číšníka, který předává přání zákazníka kuchařovi. K implementaci API lze přistupovat třemi způsoby. [34]

3.4.1 SOAP API

SOAP neboli Simple Object Access Protokol je nejstarším prostředkem, který je určený pro volání procedur za využití XML. Hlavním cílem SOAP je usnadnit odlišným aplikacím, které jsou nasazeny v různých prostředích či v odlišných jazycích, sdílení informací mezi sebou. Jde tedy především o nástroj, který odstraňuje bariéry mezi různými platformami. V dnešním světe SOAP využívají hlavně pojišťovny a banky. [24]

3.4.2 REST API

Pojem Representational State Transfer známé pod zkratkou REST se poprvé vyskytl v disertační práci Roye Fieldinga. Jedná se o architektonický styl pro navrhování distribučních systémů. Lze ho chápat spíše jako soubor omezení, nikoliv jako standard. HTTP API, které je nejčastěji využíváno s REST, je označováno jako CRUD. Jedná se o tvorbu dat (Create), která se volá metodou POST, získání dat (Retrieve) je možné docílit metodou GET, změna dat (Update) metodou PUT a smazání dat (Delete) metodou DELETE. [35]

3.4.3 GRAPHQL API

GRAPHQL je následníkem REST API. Byl vyvinut společností Meta Platforms, který ho dodnes používá v rámci své sociální sítě Facebook. V současné době se jedná o open-source protokol a díky tomu tak již může být bezplatně využíván veřejností. Umožňuje stahovat data z mnoha zdrojů současně, tento fakt výrazně ulehčuje práci programátorům. Oproti REST API má větší rychlost zpracování dat a bohatší škálu využití.

Odpadá závislost na architektuře a je možné ho implementovat nad stávajícím REST API. [34]

3.5 Low-code

Pojem low-code nebo také známé jako no-code je poměrně nový způsob vývoje aplikací, kde se místo klasického kódování používá grafické rozhraní. Platformy, které využívají tento typ vývoje se velmi rychle rozšířily, vzhledem k nedostatku kvalifikovaných vývojářů a potřebě zlepšit dobu zpracování projektů.

Využití low-code vývoje aplikací umožňuje vyvíjet aplikace mnohem rychleji než při klasickém vývoji, protože snižuje potřebu kódu. Díky tomu mohou vyvíjet jak webové, tak i mobilní aplikace i méně zkušení vývojáři, a to za využití již vyvinutých komponent. Vyvinout aplikaci pomocí low-code nástroje může i člověk, který ani není vývojářem, a tedy který nemá zkušenosti s klasickým vývojem aplikací, ale například mnohem lépe zná a rozumí obchodním procesům společnosti. Díky tomu ví, jak by tyto obchodní procesy bylo možné zlepšit či zrychlit a za pomocí low-code vývoje může snadno vyvinout potřebné řešení (aplikaci). [36]

3.6 Mendix

Mendix je vysoce produktivní aplikační platforma, která je navržena tak, aby poskytovala možnost vytvářet a vylepšovat mobilní i webové aplikace v globálním měřítku. Platforma dále urychluje poskytování podnikových aplikací, a to v průběhu celého životního cyklu vývoje aplikace, od nápadu po nasazení až po provoz. Mendix umožňuje implementovat osvědčené postupy Agile i DevOps a nabízí nástroje low-code i no-code v jediné, plně integrované platformě. Výsledkem kombinace těchto nástrojů je, že experti na obchodní domény, jakou jsou například analytici, mohou spolupracovat se zkušenými programátory, aby dosáhli rychlejšího vzájemného porozumění a tím urychlili celý proces vývoje. Cloudová nativní architektura a automatizační nástroje platformy Mendix podporují nasazení, správu a monitorování podnikových aplikací. [37]

3.6.1 Historie

Společnost Mendix byla založena v roce 2005 v Rotterdamu v Nizozemsku. Cílem této společnosti bylo od počátku vyvinout platformu, která by pomohla podnikům rychleji

dosáhnout úspěchu. 1. října 2018 společnost Mendix koupila společnost Siemens, největší průmyslová výrobní společnost v Evropě. [36]

3.6.2 Mendix App Store

Jedná se o místo, kde je možné nalézt a stáhnout moduly (funkce a konektory) nebo widgety (front-end komponenty) přímo do vyvíjené aplikace. Dále je možné na tomto místě nalézt stovky aplikací, které byly vytvořeny jak Mendixem, tak i třetími stranami. Stovky vývojářů sdílejí vytvořené widgety a moduly, které dosud nebyly vytvořeny. Díky tomu je mnohem snazší vyvíjet i složitější aplikace. [36]

3.6.3 Mendix studio

Mendix studio je webový nástroj, který umožňuje vytvářet plně funkční aplikace. S využitím Mendix studio je možné vyvíjet aplikaci bez nutnosti zacházení do větších podrobností využívaných funkcí, které nabízí Studio Pro. Mendix studio nabízí možnost spolupracovat se všemi členy týmu zároveň na jednom projektu. Největší výhodou použití aplikace je to, že lze spustit v prohlížeči a není tak nutné instalovat další software. To umožňuje vývoj aplikace kdekoliv a téměř na jakémkoliv zařízení. [36]

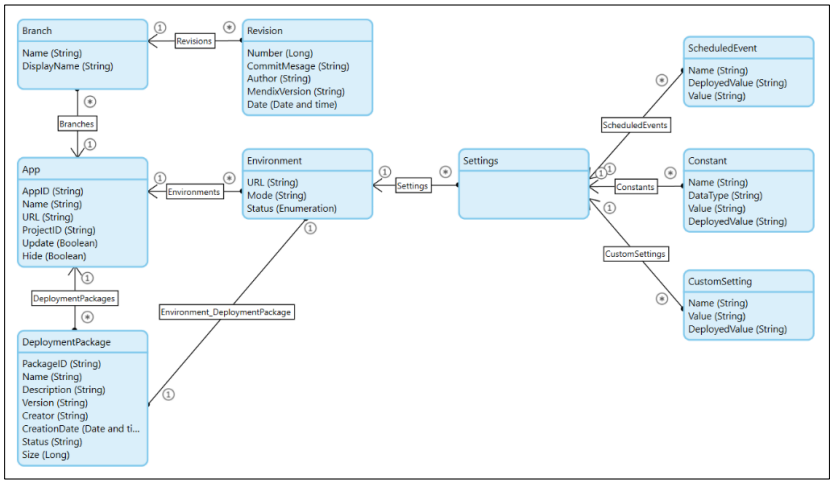
3.6.4 Studio Pro

Stejně jako Mendix studio, tak i Studio Pro je výkonný nástroj pro tvorbu aplikací. Studio Pro poskytuje možnost vytvářet robustní a komplexní aplikace. Vývojáři mohou modelovat aplikace vytvářením stránek a přidáváním logiky a konfigurací. Dále je možné testovat logiku a nasazení vyvinutých aplikací. Další výhodou je správa verzí, díky které je možné provádět vývoj na více vývojových liniích a v případě potřeby se vrátit na předchozí verzi. [36]

3.6.5 Domain Model

Domain Model neboli doménový model představuje datovou vrstvu aplikace. Když je aplikace ve vývoji, je potřeba ukládat data aplikace v nějaké struktuře. Jak Mendix Studio, tak Studio Pro poskytují nástroje potřebné pro tvorbu návrhů datových struktur aplikace a zobrazení dat v aplikaci koncovým uživatelům. Doménový model se skládá z entit (označované také jako objekty), které lze přirovnat k tabulkám v relační databázi. Tyto objekty mohou být jak perzistentní (uchovávané v databázi), tak neperzistentní (neuchovávají se v databázi). Doménový model dále obsahuje atributy, což jsou sloupce

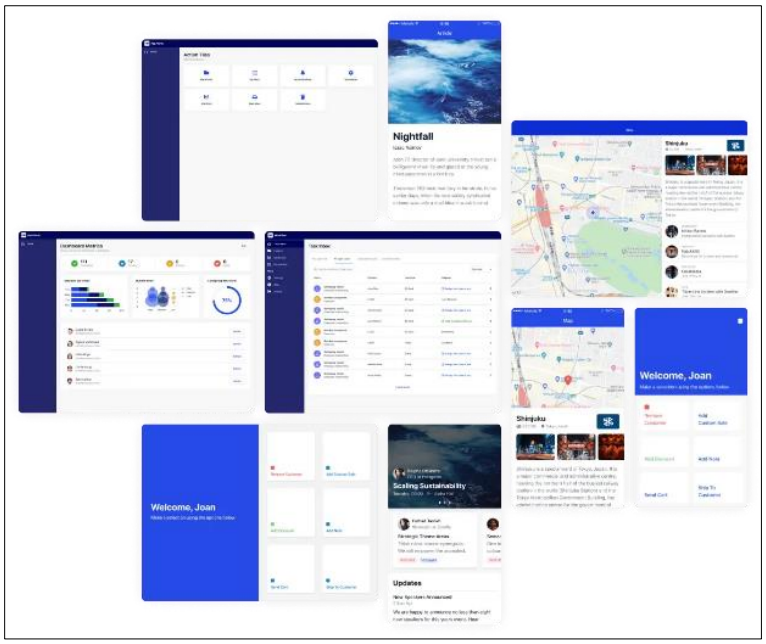
s různými datovými typy pro uložení dat. Struktura dat v doménovém modelu obsahuje asociace, které zobrazují, jak spolu jednotlivé datové objekty souvisí. [36]



Obrázek č. 12 – Domain Model [38]

3.6.6 Page design

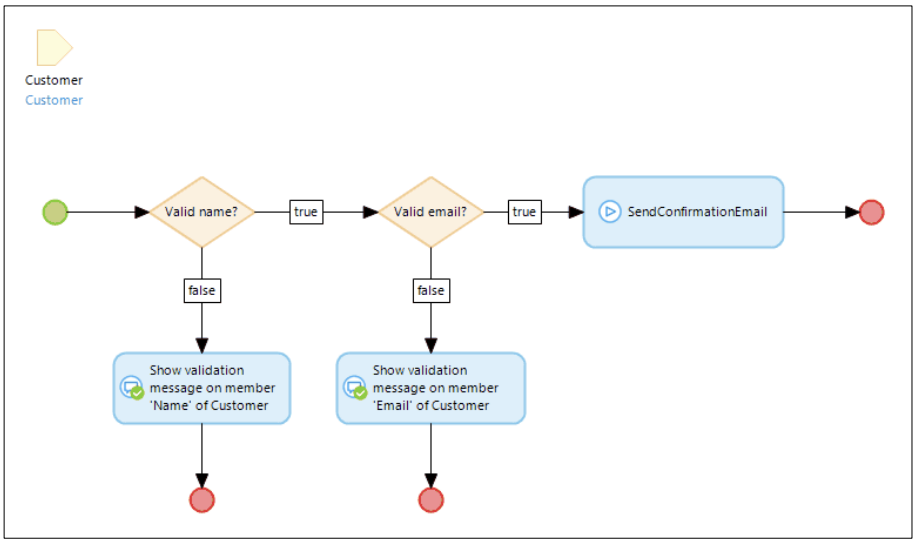
Sekce Page design umožňuje navrhnout uživatelské rozhraní aplikace. Jednotlivé stránky jsou vytvořeny pomocí návrhových rozvržení, widgetů a atlas UI ve Studio Pro. Návrhové rozvržení je možné opakovaně použít na jednotlivé stránky aplikace. Widgety poskytují prvky jako jsou textová pole, posuvníky a mnohem více. Díky tomu je uživatelům umožněna interakce se stránkami aplikace. Atlas UI poskytuje šablony stránek a další stavební bloky, které aplikaci dodají moderní design se standardními webovými prvky. [36]



Obrázek č. 13 – Atlas UI [39]

3.6.7 Microflows

K prezentační a datové vrstvě, které jsou utvořeny pomocí Page design a Domain model se ještě připojuje vrstva aplikační. Ačkoliv Mendix Studio a Studio Pro poskytují mnoho přednastavitelných tlačítek, které nabízejí okamžitou funkčnost, pro rozšíření o vlastní logiku je potřeba využití takzvané Microflows. Microflows je vizuální reprezentace programového kódu. Namísto klasického psaní kódu se využívají vizuální diagramy, díky kterým lze získávat a upravovat data, vytvářet a používat proměnné a manipulovat se stránkami aplikace. [36]



Obrázek č. 14 – Microflows [40]

3.7 Cloud Foundry

Jedná se o open-sourcovou, multicloudovou aplikaci, která byla vyvinuta společností VMware. Architektura Cloud Foundry je založena na kontejnerech a je umožněno spouštět aplikace v libovolném programovacím jazyce napříč různými poskytovateli cloudových služeb. Díky tomu mají vývojáři možnost maximálně využít cloudovou platformu, která se nejlépe hodí pro jejich aplikaci.

Hlavní výhodou Cloud Foundry je již zmíněná kompatibilita napříč programovacími jazyky. Je tedy možné pracovat s programovacími jazyky jako jsou Java, PHP, Python nebo Ruby. Nevýhodou jsou však linuxové kontejnery, které jsou oproti konkurenčním řešením, jako je například OpenShift nebo Red Hat, stále pozadu. [41]

3.8 HTML a kaskádové styly CSS

HyperText Markup Language, zkráceně HTML je základní technologie, ve které jsou vytvořeny všechny webové stránky. HTML je jazyk značkovací nikoliv programovací, jelikož k popisu webových stránek používá značky, tzv. tagy. [45]

Samotné HTML se používá ke strukturování webu (definování prvků, jako jsou nadpisy a odstavce, umožnění vkládání obrázků, videí a dalších médií). Naproti tomu CSS (kaskádové styly) určují styl webu (rozvržení stránky, barvy, písma a velikosti). CSS dodává styl webu pomocí interakcí s prvky z kódu HTML.

Existují 3 způsoby, jak propojit CSS s HTML. Kód CSS může být externí, tedy uložen zvlášť (mimo HTML) jako soubor CSS a má možnost určovat vzhled celého webu prostřednictvím jednoho souboru. Další způsob propojení je interní, jedná se o CSS zapsané přímo do záhlaví konkrétní stránky HTML. Využití interního vložení CSS je vhodné, pokud je na webu daná stránka, která má jiný vzhled než ostatní. Poslední možností propojení CSS a HTML je skrze vložení stylů jako fragmenty CSS napsané přímo v HTML, které je možné použít pouze na jeden prvek kódu HTML. [46]

3.9 Drátěný model

Drátěný model, také znám jako wireframe je podstatným prvkem v procesu tvorby webové stránky. Drátěný model určuje, co bude web obsahovat (jaké stránky), jak web bude fungovat, ale také jaký bude obsah jednotlivých stránek a jejich propojení. Správně vypracovaný drátěný model také určuje rozvržení a počet obrázků nebo délku obsahových textů. [48]

3.10 Životní cyklus aplikace

Aplikace prochází jednotlivými fázemi vývoje, až se po určité době vrátí znovu na začátek. Proto se celý tento proces nazývá životní cyklus aplikace. V určitém okamžiku, kdy se aplikace vrátí na začátek, musí být připravena na vyšší úroveň pomocí nových technologií vycházejících z jiných potřeb uživatele.

Tento cyklus začíná plánováním a přípravou aplikace. Následně je provedena analýza a návrh celé aplikace. Další fází je samotná implementace aplikace, po které následuje příprava pro zavedení do provozu. Po přípravě je další fází provoz a údržba a poslední fází je pak další rozvoj a optimalizace. [47]

3.10.1 Plánování a příprava

Každá nová aplikace vychází z informační strategie rozvoje podniku a požadavků uživatelů. Na samotném začátku této fáze je pouhý záměr aplikaci řešit, avšak na konci dané fáze musí být jednoznačné, zda se aplikace bude realizovat a případně s jakými cíli, funkcionalitou atd. Do plánování a přípravy spadá i prvotní analýza nebo také zpracování projektového záměru projektu, na jehož základě je následně vybírán dodavatel projektu aplikace. [47]

3.10.2 Analýza a návrh aplikace

V této fázi je nutné dále definovat požadavky na aplikaci z hlediska funkcí, které by měla poskytovat. Kromě toho, která data zpracovávat a které obchodní procesy by měla aplikace podporovat. Tato sada funkcí je v praxi strukturována odlišně v závislosti jak na typu metody návrhu, tak na typu aplikace. Analýzu v této fázi je možné rozdělit na analýzu podnikových procesů, analýzu stávajících databází a analýzu stávajících aplikací.

Návrh lze rozdělit do tří hlavních celků, a to na návrh změn podnikových procesů, kde je možné definovat nové podnikové procesy, které má aplikace podporovat. Následně na návrh databází, které se skládají z návrhu dat, datových bází a jejich obsahu. A na návrh aplikace, kde jsou cílová řešení aplikace rozdělována na dvě základní úrovně, a to na fyzickou a logickou. [47]

3.10.3 Implementace aplikace

V této fázi implementace zahrnuje přesné definování jednotlivých programových modulů, ze kterých se následně vytváří prototypy, které důkladněji prověří skutečné potřeby uživatele. Díky tomu jsou snížena rizika omylů při formulaci funkcionalit aplikace. Dále je podle konkrétního řešení provedena kastomizace funkcí, vývoj nebo dovývoj specializovaných programových modulů. Následně je provedeno akceptační řízení, které je z těchto dílčích částí poslední. Aplikace se testuje s testovacími daty odpovídajícími reálné situaci informačního sytému zákazníka. Ve všech těchto fázích implementace je potřeba provádět důkladnou dokumentaci. [47]

3.10.4 Příprava na zavedení do provozu

Existují dva způsoby, kterými je možné provést zavedení aplikace do provozu. Jednou z možností je ukončit dosavadně nasazenou aplikaci a nasadit aplikaci novou.

Druhou možností je postupný přechod na novou aplikaci, přičemž jsou obě aplikace po určitou dobu provozovány společně a dochází k porovnání výsledků. První možnost snižuje zatížení pracovníků s provozem obou aplikací, ale naopak se zvyšuje riziko dopadů případných chyb aplikace. Obě varianty je potřeba pečlivě zvážit. Hlavními kritérii jsou situace podniku, charakter podniku a význam aplikace. [47]

3.10.5 Provoz a údržba

Tato fáze zahrnuje běžné údržbové operace, provozní servis a permanentní konzultační služby, tedy podpora. Podpora je velice důležitým zdrojem pro další rozvoj, jelikož kromě operativního řešení problémů, zahrnuje tato služba i evidenci a vyhodnocování požadavků uživatelů. Dalším nezbytným zdrojem pro další rozvoj aplikace je monitorování, kdy se zpracovávají provozní statistiky, které zahrnují monitorování poruch a provozních chyb technologií, na nichž je aplikace provozována. [47]

3.10.6 Rozvoj a optimalizace aplikace

Rozvoj aplikace a její optimalizace má charakter průběžných úprav nebo zásadní změny celého řešení. Vstupem do této fáze je analýza existujících nových požadavků na aplikaci v rámci tzv. změnového řízení. Následně je určeno, zda se bude jednat o dílčí změnu aplikace, která bude realizována v rámci údržby, nebo o zásadní změnu pro kterou je nutné provést specifikaci nového projektu. V obou z těchto variant je nutné provést analýzu a definovat nově požadované funkcionality, vstupní a výstupní datové struktury, případně návrhy změn technologické infrastruktury a provozu aplikace. [47]

3.11 Metodiky vývoje softwaru

Metodiky vývoje softwaru lze členit na dvě hlavní metodiky, a to jsou rigorózní metodiky zvané také jako těžké metodiky a agilní metodiky, které se také nazývají jako lehké metodiky. Obě tyto metodiky mají různorodý pohled na vývoj softwaru a vychází z odlišných předpokladů. Váha metodiky se považuje za jednu z hlavních kritérií, která odlišuje tyto dvě metodiky. Rigorózní metodiky vznikly z předpokladu, že budování IS/ICT je možné popsat, plánovat, řídit a měřit. Podrobně a přesně definují procesy, vytvářené produkty a činnosti, a z tohoto důvodů bývají často objemné. Oproti tomu agilní metodiky jsou projektové metody, které necílí na údržbu a provoz, ale na vývoj nového řešení. Agilní

metodiky umožňují vytvořit velice rychlé řešení, které je možné dynamicky přizpůsobovat měnícím se požadavkům. [42]

3.11.1 Kategorizace metodik

Rigorózní metodiky je možné rozdělit na:

- **Metodika OPEN** – primárně se zaměřuje na vývoj komponentových a objektově orientovaných aplikací. Podporuje celý životní cyklus vývoje IS/ICT a je veřejně dostupná.
- **Metodika Rational Unified Process (RUP)** – je založena softwarových praktikách vývoje jako jsou iterativní vývoj, řízení požadavků, vizuální modelování, kontrola kvality softwaru, použití komponentové architektury a řízení změn.
- **Metodika Enterprise Unified Process (EUP)** – rozšiřuje metodiku RUP ve dvou směrech. První směr představuje rozšíření na úroveň celé organizace a druhý směr cílí na provoz a údržbu systému a také popisuje úkony, které jsou nutné provést při odstranění produktu z používání.
- **Metodika MMDIS –** tato metodika je zaměřena na architekturu budování informačního systému a systémovou integraci. Je zaměřena na celou organizaci a spadá tak do kategorie globálních metodik.

Agilní metodiky je možné rozdělit na:

- **Dynamic Systém Development Method (MSDM)** – metodika je postavena na 9 principech a představuje rozšíření praktik rychlého vývoje aplikací. Ze všech agilních metodik má MSDM nejkvalitnější systém školení a dokumentace.
- **Adaptive Software Development (ASD)** – využívá dynamický životní cyklus, jehož základem je kontinuální učení a hybnou silou jsou neustálé změny. Životní cyklus ASD se skládá ze tří částí, které se nazývají spekulace, spolupráce a učení.
- **Lean Development** – založena na principu dynamické stability, tedy schopnosti přizpůsobení se rychle a efektivně požadavkům. Díky tomu jsou vytvářeny stabilní a permanentně zlepšující se vnitřní procesy.
- **Feature-Driven Development (FDD)** – využívá iterativní vývoj, který je řízen užitnými vlastnostmi produktu. Prvotně je vytvořen model, který

pokračuje posloupností dvoutýdenních iterací, ve kterých je prováděn návrh a realizace pro jednotlivé užitné vlastnosti. Užitná vlastnost je malý výsledek, který je užitečný z pohledu zákazníka.

- **Crystal metodiky** – jedná se o soubor metod, jejichž silnou stránkou je komunikace a lehkost produktu. Jednotlivé prvky metodiky jsou přizpůsobeny individuálně pro všechny projekty. Výběr vhodné metodiky z rodiny Crystal je založen na třech aspektech. První je velikost projektu. Dalším hlediskem je důležitost systému. Poslední úvaha je to, pro co je metoda optimalizována
- **Scrum –** primárně se zaměřuje na řízení projektu. Procesy vývoje softwaru jsou v metodice chápány jako empirické procesy, které nelze žádným způsobem předvídat, ale je nutné je monitorovat. Software je vyvíjen ve třicetidenních iteracích nazývaných Sprinty, které obsahují na konci množinu užitných vlastností.
- **Extrémní programování (XP)** – tato metodika je vhodná pro malé až středné velké firmy. Představuje vývoj softwaru, jehož zadání se průběžně mění nebo není zcela vydefinováno. Je kladen důraz na kvalitu technického řešení a vytváření testů před kódováním. [47]

Metodik pro budovaní IS/ICT je velké množství, které není jednotným způsobem popsáno a ani dobře kategorizováno. To má za následek znesnadnění vyhledání vhodné metodiky pro konkrétní typ projektu. Skutečnosti, které mají za příčinu existenci různých metodik jsou:

- **Různé technologie vyžadují různé techniky a metody** – například objektově orientované metodiky vyhovují projektům využívající objektově orientované technologie. Naopak datově orientované metodiky vyhovují vývoji datově orientovaných aplikací.
- **Organizace se odlišuje svou firemní kulturou –** firemní kultura může být s určitými přístupy v rozporu, a to má za následek selhání při implementaci určité metodiky. Před implementací metodiky v organizaci je potřeba nejprve provést analýzu firemní kultury.
- **Každý jedinec je výjimečný –** lidé jsou primárním faktorem při vývoji informačního systému. Metodiky tento fakt zohledňují pouze v malém měřítku. Lidé se však navzájem odlišují svými znalostmi či způsobem dosahování cílů. Z tohoto důvodu

není možné vytvořit univerzální metodiku, která by vyhovovala všem, ale naopak je třeba ji přizpůsobit konkrétním lidem, jejich znalostem a schopnostem.

- **Každý tým je jedinečný –** jedinečnost jedinců vede i k jedinečnosti týmu.
- **Projekty jsou různé dle velikostí týmu –** pro malý počet lidí je vhodné použití malé metodiky.
- **Projekty jsou odlišné svou důležitostí –** vytváření systému pro řízení letového provozu vyžaduje zcela odlišnou metodiku než například mzdová agenda.
- **Projekty jsou odlišné dle postavení produktu na trhu –** produkt, který vstupuje na trh většinou nepoužívá žádnou metodiku anebo jen velmi lehkou metodiku. Naopak na produkty již zavedené na trhu je možné aplikovat rigorózní metodiku s přesně popsanými procesy. Produkt, díky kterému organizace získá konkurenční výhodu je potřeba vyvinout rychle a je vhodné na něj naopak uplatnit agilní metodiky vývoje softwaru.
- **Projekt existuje v rámci určitého specifického vnějšího prostředí –** některé projekty musí respektovat pravidla pro státní zakázky a některé jsou vázány na dodavatele a jeho přesně stanovené požadavky. [42]

Jednotlivé metodiky se odlišují například tím, jakou část životního cyklu postihují nebo jakým způsobem definují jednotlivé kroky při budování informačního systému. Kritéria, která lze využít pro kategorizaci metodik budování IS/ICT jsou:

- Zaměření metodiky
- Rozsah metodiky
- Váha metodiky
- Typ řešení
- Doména
- Přístup k řešení [42]

3.12 Analýza existujících řešení

V platformě MindSphere existují dvě možnosti zasílání notifikací koncovým zákazníkům. První z možností je skrze aplikaci SIMATIC Notifier a druhá možnost je skrze systémovou aplikaci Fleet Manager.

SIMATIC Notifier je aplikace, ve které je možné vytvářet pravidla nad parametry, které je potřeba sledovat vůči svému vychýlení. Ve chvíli, kdy je aktivované pravidlo, zašle

oznámení koncovému zákazníkovi do mobilní aplikace, nebo je možné vidět oznámení přímo ve webovém uživatelském rozhraní aplikace. [43]

Fleet Manager je aplikace, které má v sobě zakomponováno více funkcí, než je pouze vytváření notifikací. Dále je možné v aplikaci sledovat time series data, exportovat data a také například zobrazit polohu zařízení. Fleet Manager u aktivovaného pravidla vygeneruje událost (event), kterou je možné vidět v time series datech, event mangementu nebo má také možnost zasílat notifikace koncovým zákazníkům skrze e-mail. [44]

3.13 Zhodnocení existujících řešení

Jak SIMATIC Notifier, tak Fleet Manager mají možnost vytvářet pravidla, velmi snadno si označit sledovaný parametr a také vytvářet logiku pravidla. Dále je umožněno zasílat notifikace vybraným zákazníkům a je částečně umožněné ovlivnit obsah zprávy. Koncový zákazník ani uživatel těchto aplikací ale nemá žádnou možnost ovlivnit, jak daná notifikace vypadá či v jakém je jazyce. Jediné, co je možné v notifikaci ovlivnit je popis aktivovaného pravidla, nikoliv vzhled či ostatní zobrazované parametry. Ve chvíli, kdy koncový zákazník nerozumí anglicky (jak je standardně notifikace zasílána), pak mu tato notifikace nepřinese žádnou informaci a stává se tak neužitečná. To samé platí i o nemožnosti ovlivnit vzhled notifikace či ovlivnění obsahu notifikace. Vzhledem k těmto hrubým nedostatkům u jednotlivých řešeních, lze tyto řešení označit za nedostatečné.

4 Vlastní práce

4.1 Stanovení cílů aplikace

Cílem této notifikační aplikace je možnost ovlivnit vzhled zasílaných notifikací, jejich obsah a možnost ovlivnit komu se notifikace budou zasílat. Aplikace proto musí splňovat tyto body:

- Přehled všech vytvořených notifikací
- Přehled všech vytvořených pravidel
- Přehled všech importovaných šablon
- Přehled všech vytvořených kontaktů
- Přidání nové notifikace
- Přidání nového kontaktu administrátorem
- Vytváření e-mailové šablony
- Možnost importu e-mailové šablony
- Upravování vytvořené notifikace
- Upravování vytvořeného kontaktu administrátorem
- Upravování vytvořené šablony
- Možnost smazání vytvořené notifikace
- Možnost smazání vytvořeného kontaktu administrátorem
- Možnost smazání vytvořené šablony

4.2 Omezení práce s aplikací

Pro využívání aplikace je aplikace rozdělena do dvou vrstev oprávnění. Tyto vrstvy se vzájemně odhlašují podle toho, zda aplikaci využívá uživatel se standardním oprávněním nebo oprávněním typu administrátor.

První vrstva je určena pro jakéhokoliv uživatele OS MindSphere, který nemá přiřazené oprávnění typu administrátor pro notifikační aplikaci. Má možnost vytvářet, upravovat, mazat notifikace, vytvářet, importovat, upravovat, mazat šablony. Dále má umožněno zobrazovat pravidla a zobrazovat kontakty.

Druhá vrstva je určena pro jakéhokoliv uživatele OS MindSphere, který má přiřazené oprávnění typu administrátor pro notifikační aplikaci. Oproti standardnímu uživatelskému

oprávnění má administrátor možnost přidávání a upravování kontaktů, kterým se notifikace budou zasílat.

4.3 Use Case specifikace

Obsahem této specifikace jsou jednotlivé případy užití tzv. Use Cases. Jedná se o metodu popisu sekvence kroků, které vykonává uživatel při plnění konkrétního úkolu za použití interagujícího systému. Specifikace obsahuje případy užití pro potvrzení zaslané notifikace, přidání šablony, úpravu šablony, smazání šablony, přidání notifikace, úpravu notifikace, smazání notifikace, přidání kontaktu, úpravu kontaktu, smazání kontaktu a úpravu pravidla.

Název	UC01 – Potvrzení zaslané notifikace
Popis	Potvrzení zaslané notifikace
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu uživatel nebo administrátor a nachází se na příslušné stránce pro možnost potvrzení zaslané notifikace
Základní tok	1. Systém zobrazí 6 posledních zaslaných notifikací 2. Uživatel vybere zaslanou notifikaci, potvrdí zaškrtávací políčko a stiskne tlačítko „Acknowledge“ 3. Systém nastaví parametr acknowledge vybrané zaslané notifikace na hodnotu true 4. Systém skryje potvrzenou notifikaci
Alternativní tok 1	1.1 Pokud nebyla zaslaná žádná notifikace, systém upozorní uživatele
Podmínky dokončení	Systém nastaví parametr vybrané zaslané notifikace na hodnotu true a danou notifikaci skryje

Tabulka 1 – Případ užití UC01 – Potvrzení zaslané notifikace

Název	UC02 – Přidání šablony
Popis	Umožňuje přidat novou šablonu
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu uživatel nebo administrátor a nachází se na příslušné stránce pro přidání šablony
Základní tok	1. Systém vygeneruje formulář 2. Uživatel vyplní název, popis, obsah šablony a stiskne tlačítko „Save“ 3. Systém provede validaci dat od uživatele 4. Systém uloží šablonu
Alternativní tok 1	3.1 Pokud uživatel zadal neplaný vstup nebo vstupy, systém upozorní uživatele a neuloží šablonu do databáze 3.2 Uživatel opraví neplatný vstup nebo vstupy a tok pokračuje na 2. bodu základního toku
Alternativní tok 2	2.1 Uživatel vyplní název, popis a pro vložení obsahu šablony stiskne tlačítko „Select“ 2.2 Systém zobrazí adresářovou strukturu zařízení 2.3 Uživatel vybere soubor pro obsah šablony 2.4 Systém načte obsah souboru do obsahu šablony 2.5 Uživatel stiskne tlačítko „Select“ a tok pokračuje na 3. bodu základního toku
Podmínky dokončení	Systém korektně uloží šablonu do databáze šablon

Tabulka 2 – Případ užití UC02 – Přidání šablony

Název	UC03 – Úprava šablony
Popis	Umožňuje upravit šablonu
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu uživatel nebo administrátor a nachází se na příslušné stránce pro zobrazení všech šablon
Základní tok	1. Systém načte a zobrazí všechny šablony z databáze 2. Uživatel vybere šablonu 3. Systém zobrazí detail zvolené šablony 4. Uživatel stiskne tlačítko „Edit“ 5. Systém zobrazí stránku pro úpravu šablony 6. Uživatel upraví šablonu a stiskne tlačítko „Save“ 7. Systém provede validaci dat od uživatele 8. Systém uloží změny v šabloně
Alternativní tok 1	7.1 Pokud uživatel zadal neplaný vstup nebo vstupy, systém upozorní uživatele a neuloží změny šablony do databáze 7.2 Uživatel opraví neplatný vstup nebo vstupy a tok pokračuje na 6. bodu základního toku
Podmínky dokončení	Systém korektně uloží změny upravované šablony do databáze

Tabulka 3 – Případ užití UC03 – Úprava šablony

Název	UC04 – Smazání šablony
Popis	Umožňuje smazat šablonu
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu uživatel nebo administrátor a nachází se na příslušné stránce pro zobrazení všech šablon
Základní tok	1. Systém načte a zobrazí všechny šablony z databáze 2. Uživatel vybere šablonu 3. Systém zobrazí detail zvolené šablony 4. Uživatel stiskne tlačítko „Delete“ 5. Systém zobrazí pop-up okno s potvrzením smazání 6. Uživatel stiskne tlačítko „OK“ 7. Systém vymaže zvolenou šablonu z databáze
Podmínky dokončení	Systém korektně vymaže zvolenou šablonu z databáze šablon

Tabulka 4 – Případ užití UC04 – Smazání šablony

Název	UC05 – Přidání notifikace
Popis	Umožňuje přidat novou notifikaci
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu uživatel nebo administrátor a nachází se na příslušné stránce pro přidání notifikace
Základní tok	1. Systém vygeneruje formulář 2. Uživatel vyplní uživatelský název, popis a stiskne tlačítko „Next“ 3. Systém načte kontakty a zobrazí druhou část formuláře 4. Uživatel vybere kontakt a stiskne tlačítko „Next“ 5. Systém zobrazí třetí část formuláře pro přidání assetu a pravidla 6. Uživatel stiskne tlačítko „Assign an asset and rule“ 7. Systém pošle API dotaz na všechny assety do MindSphere a zobrazí je 8. Uživatel vybere asset 9. Systém pošle API dotaz na pravidla zvoleného assetu 10. Uživatel vybere pravidlo a stiskne tlačítko „Done“ 11. Systém zobrazí třetí část formuláře s vyplněným polem assetu i pravidla 12. Uživatel stiskne tlačítko „Next“ 13. Systém načte šablony a zobrazí poslední část formuláře 14. Uživatel vybere šablonu a stiskne tlačítko „Save“ 15. Systém provede validaci dat od uživatele 16. Systém uloží notifikaci
Alternativní tok 1	15.1 Pokud uživatel zadal neplaný vstup nebo vstupy, systém upozorní uživatele a neuloží notifikaci do databáze. 15.2 Uživatel opraví neplatný vstup nebo vstupy a tok pokračuje na 14. bodu základního toku
Podmínky dokončení	Systém korektně uloží notifikaci do databáze notifikací

Tabulka 5 – Případ užití UC05 – Přidání notifikace

Název	UC06 – Úprava notifikace
Popis	Umožňuje upravit notifikaci
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu uživatel nebo administrátor a nachází se na příslušné stránce pro zobrazení všech notifikací
Základní tok	1. Systém načte a zobrazí všechny notifikace z databáze 2. Uživatel vybere notifikaci 3. Systém zobrazí detail zvolené notifikace 4. Uživatel stiskne tlačítko „Edit“ 5. Systém zobrazí stránku pro úpravu notifikace 6. Uživatel upraví notifikaci a stiskne tlačítko „Save“ 7. Systém provede validaci dat od uživatele 8. Systém uloží změny v notifikaci
Alternativní tok 1	7.1 Pokud uživatel zadal neplaný vstup nebo vstupy, systém upozorní uživatele a neuloží změny notifikace do databáze 7.2 Uživatel opraví neplatný vstup nebo vstupy a tok pokračuje na 6. bodu základního toku
Podmínky dokončení	Systém korektně uloží změny upravované notifikace do databáze

Tabulka 6 – Případ užití UC06 – Úprava notifikace

Název	UC07 – Smazání notifikace
Popis	Umožňuje smazat notifikaci
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu uživatel nebo administrátor a nachází se na příslušné stránce pro zobrazení všech notifikací
Základní tok	1. Systém načte a zobrazí všechny notifikace z databáze 2. Uživatel vybere notifikaci 3. Systém zobrazí detail zvolené notifikace 4. Uživatel stiskne tlačítko „Delete“ 5. Systém zobrazí pop-up okno s potvrzením smazání 6. Uživatel stiskne tlačítko „OK“ 7. Systém vymaže zvolenou notifikaci z databáze
Podmínky dokončení	Systém korektně vymaže zvolenou notifikaci z databáze notifikací

Tabulka 7 – Případ užití UC07 – Smazání notifikace

Název	UC08 – Přidání kontaktu
Popis	Umožňuje přidat nový kontakt
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu administrátor a nachází se na příslušné stránce pro přidání kontaktu
Základní tok	1. Systém vygeneruje formulář 2. Uživatel vyplní e-mailovou adresu, jméno, příjmení, popis a stiskne tlačítko „Save“ 3. Systém provede validaci dat od uživatele 4. Systém uloží kontakt
Alternativní tok 1	3.1 Pokud uživatel zadal neplaný vstup nebo vstupy, systém upozorní uživatele a neuloží kontakt do databáze 3.2 Uživatel opraví neplatný vstup nebo vstupy a tok pokračuje na 2. bodu základního toku
Podmínky dokončení	Systém korektně uloží kontakt do databáze kontaktů

Tabulka 8 – Případ užití UC08 – Přidání kontaktu

Název	UC09 – Úprava kontaktu
Popis	Umožňuje upravit kontakt
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu administrátor a nachází se na příslušné stránce pro zobrazení všech kontaktů
Základní tok	1. Systém načte a zobrazí všechny kontakty z databáze 2. Uživatel vybere kontakt 3. Systém zobrazí detail zvoleného kontaktu 4. Uživatel stiskne tlačítko „Edit“ 5. Systém zobrazí stránku pro úpravu kontaktu 6. Uživatel upraví kontakt a stiskne tlačítko „Save“ 7. Systém provede validaci dat od uživatele 8. Systém uloží změny v kontaktu
Alternativní tok 1	7.1 Pokud uživatel zadal neplaný vstup nebo vstupy, systém upozorní uživatele a neuloží změny kontaktu do databáze 7.2 Uživatel opraví neplatný vstup nebo vstupy a tok pokračuje na 6. bodu základního toku
Podmínky dokončení	Systém korektně uloží změny upravovaného kontaktu do databáze

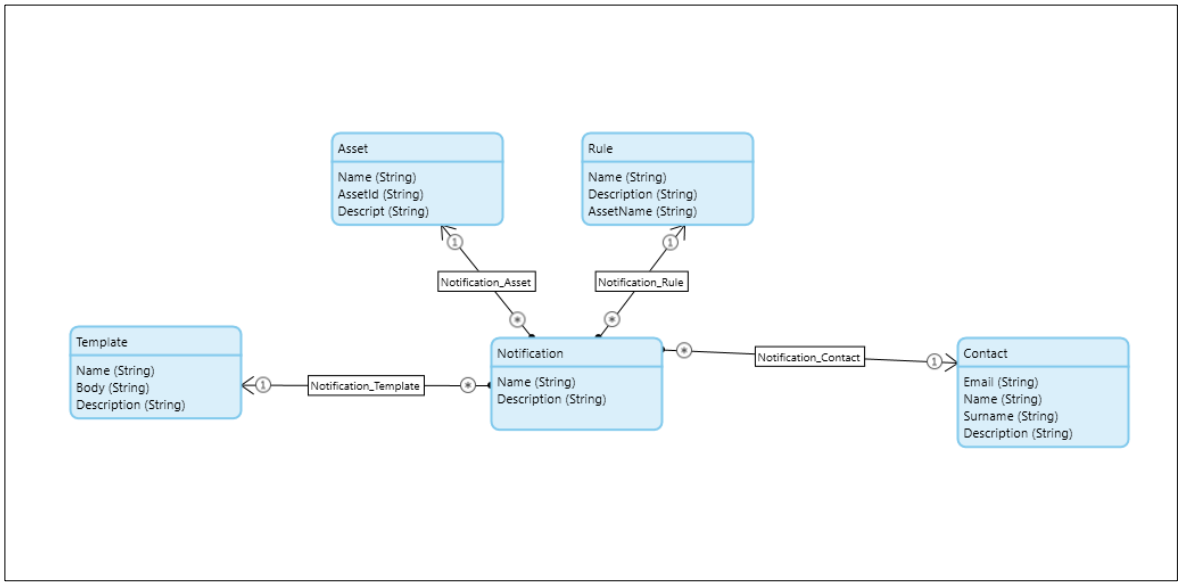
Tabulka 9 – Případ užití UC09 – Úprava kontaktu

Název	UC10 – Smazání kontaktu
Popis	Umožňuje smazat kontakt
Aktéři	Uživatel, systém
Podmínky spuštění	Přihlášený uživatel do MindSphere, který má oprávnění v aplikaci typu administrátor a nachází se na příslušné stránce pro zobrazení všech kontaktů
Základní tok	1. Systém načte a zobrazí všechny kontakty z databáze 2. Uživatel vybere kontakt 3. Systém zobrazí detail zvoleného kontaktu 4. Uživatel stiskne tlačítko „Delete“ 5. Systém zobrazí pop-up okno s potvrzením smazání 6. Uživatel stiskne tlačítko „OK“ 7. Systém vymaže zvolený kontakt z databáze
Podmínky dokončení	Systém korektně vymaže zvolený kontakt z databáze kontaktů

Tabulka 10 – Případ užití UC10 – Smazání kontaktu

4.4 Datová vrstva

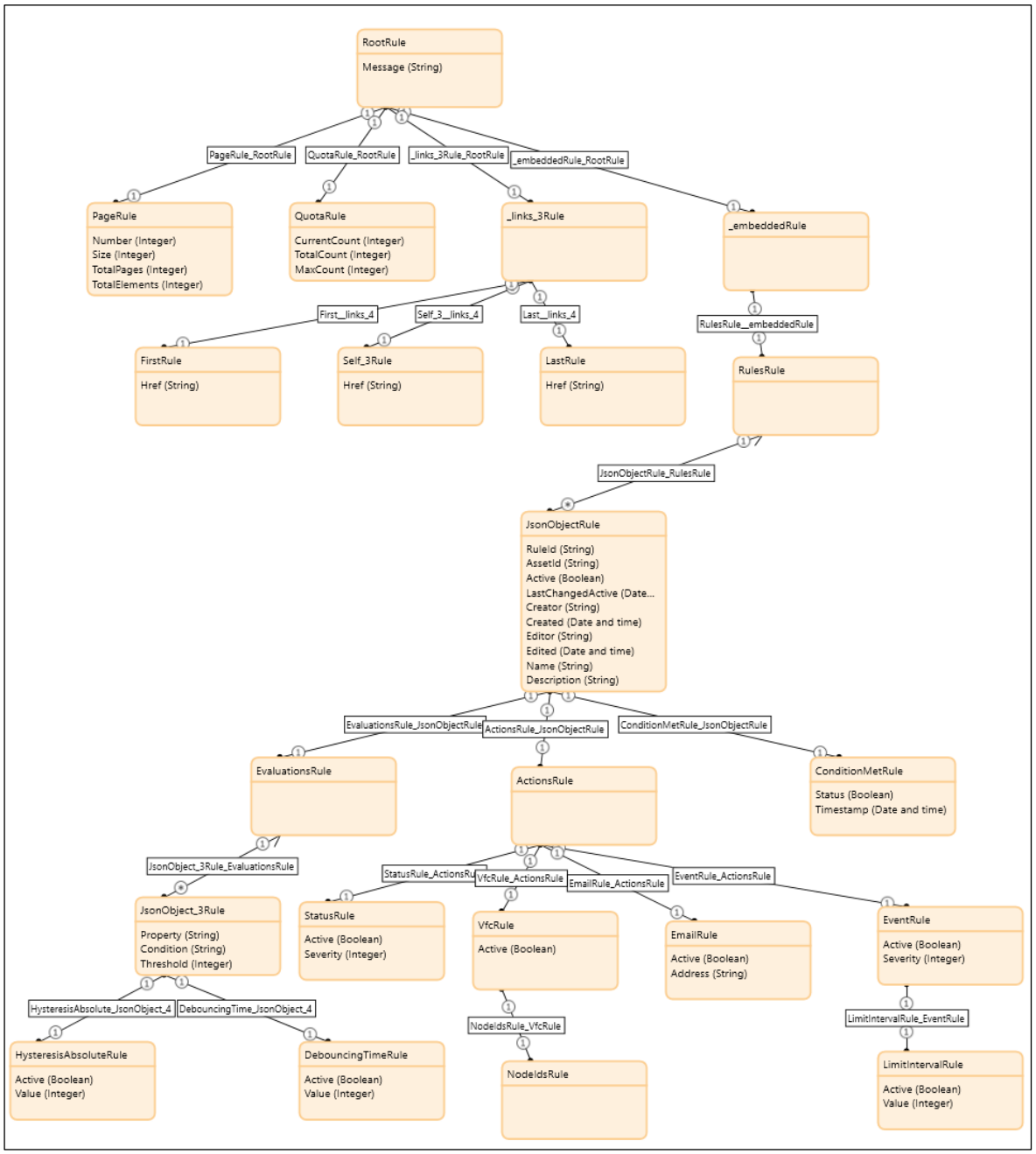
Datová vrstva je v Mendixu řešena doménovým modelem. Pro splnění všech vytyčených cílů aplikace bylo potřeba vytvořit datové struktury, a to jak pro perzistentní, tak i pro neperzistentní data. Perzistentní entity, které společně tvoří jednu vzájemně provázanou datovou strukturu jsou znázorněny na obrázku číslo 15.



Obrázek č. 15 – Perzistentní datová struktura

Neperzistentní datové struktury byly vytvořeny na základě odpovědí provolávaných API požadavků. Jedná se o provolání událostí, assetů a pravidel. Jedná se vždy o mnoho

entit s několika parametry, které jsou vzájemné provázané v jednu datovou strukturu. Datovou strukturu pro zpracování neperzistentních dat pravidel je možné vidět na obrázku číslo 16.



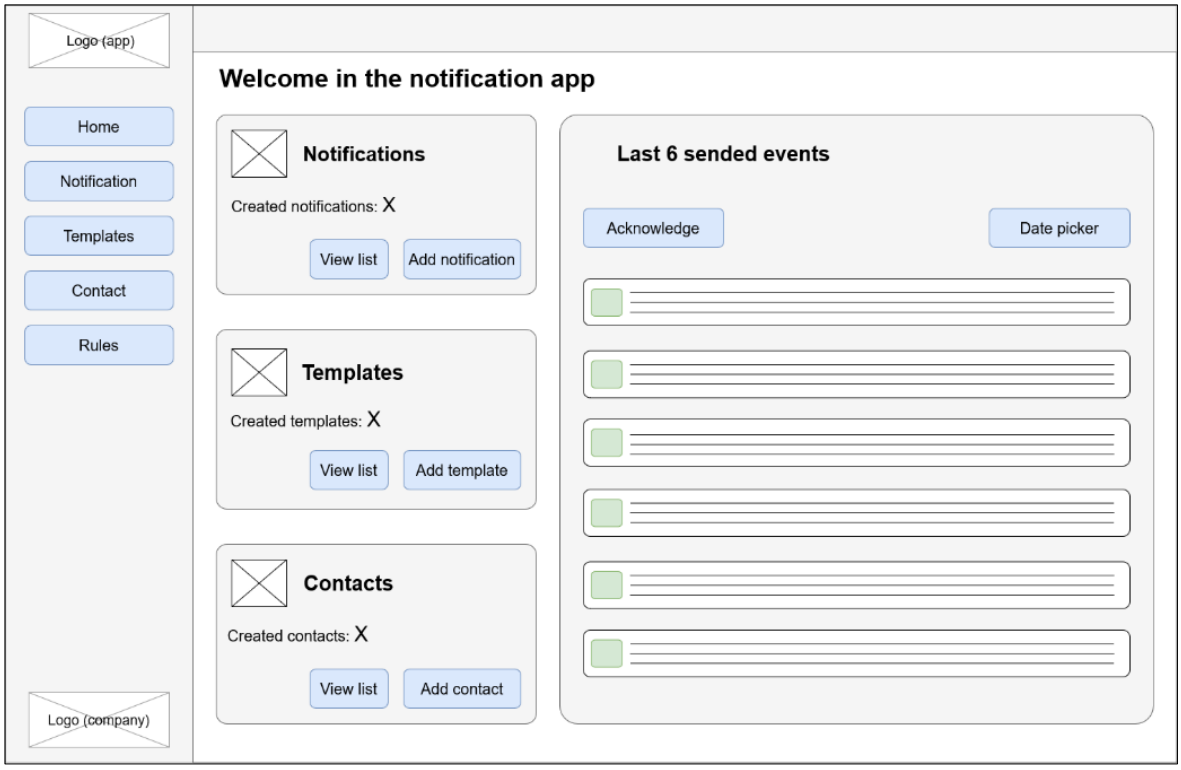
Obrázek č. 16 – Neperzistentní datová struktura pravidel

4.5 Drátěný model (wireframe)

Pro tvorbu drátěných modelů neboli wireframů byl zvolen on-line nástroj draw.io, kde je umožněno zdarma vytvářet různé typy diagramů, jako jsou vývojové diagramy nebo síťové diagramy. Digramy je možné nejen vytvářet, ale i ukládat nebo exportovat v podobě obrázků.

4.5.1 Úvodní obrazovka

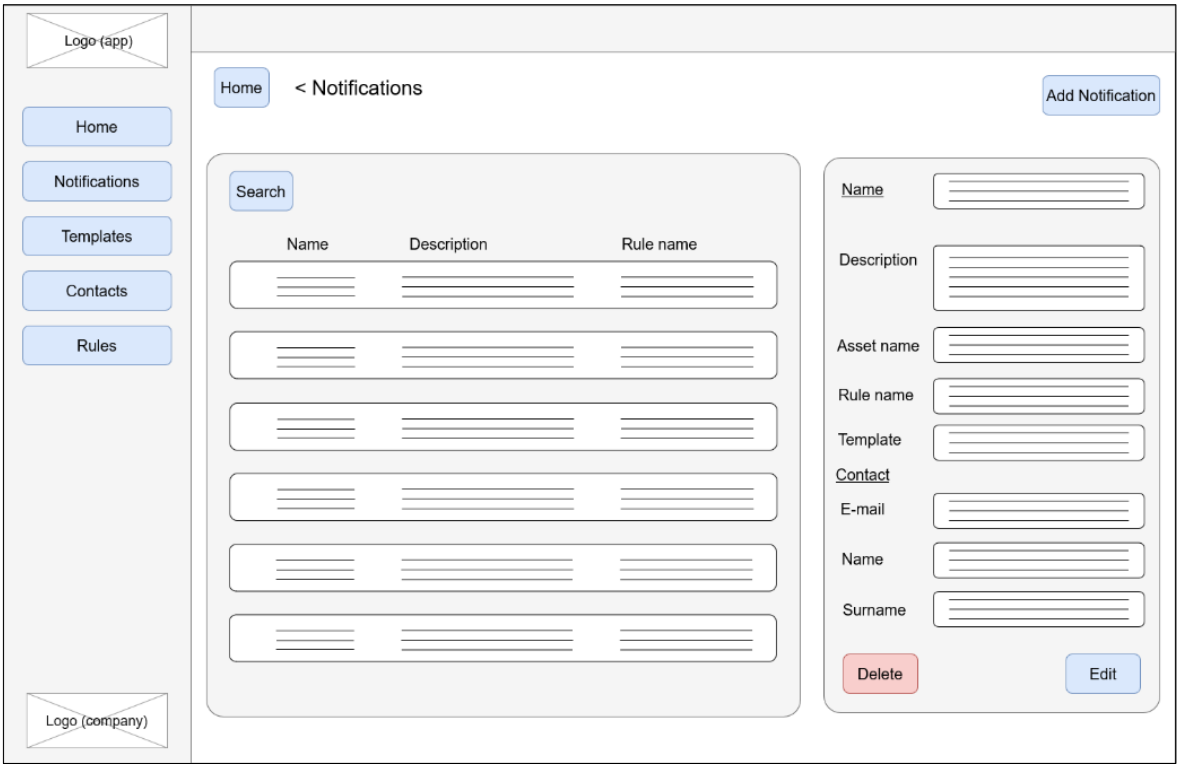
Na úvodní obrazovce má uživatel ihned přehled o nejdůležitějších parametrech aplikace. Zobrazuje se zde počet vytvořených notifikací, počet vytvořených šablon nebo počet vytvořených kontaktů. Kromě těchto parametrů má uživatel také možnost vidět posledních 6 událostí, které vznikly a které může uživatel jednoduše potvrdit, že si jich je vědom. Dále má uživatel možnost ihned přejít na stránku pro přidání notifikace, přidání šablony a pokud má administrátorské oprávnění, tak i na přidání kontaktů.



Obrázek č. 17 – Wireframe úvodní obrazovky

4.5.2 Přehled notifikací

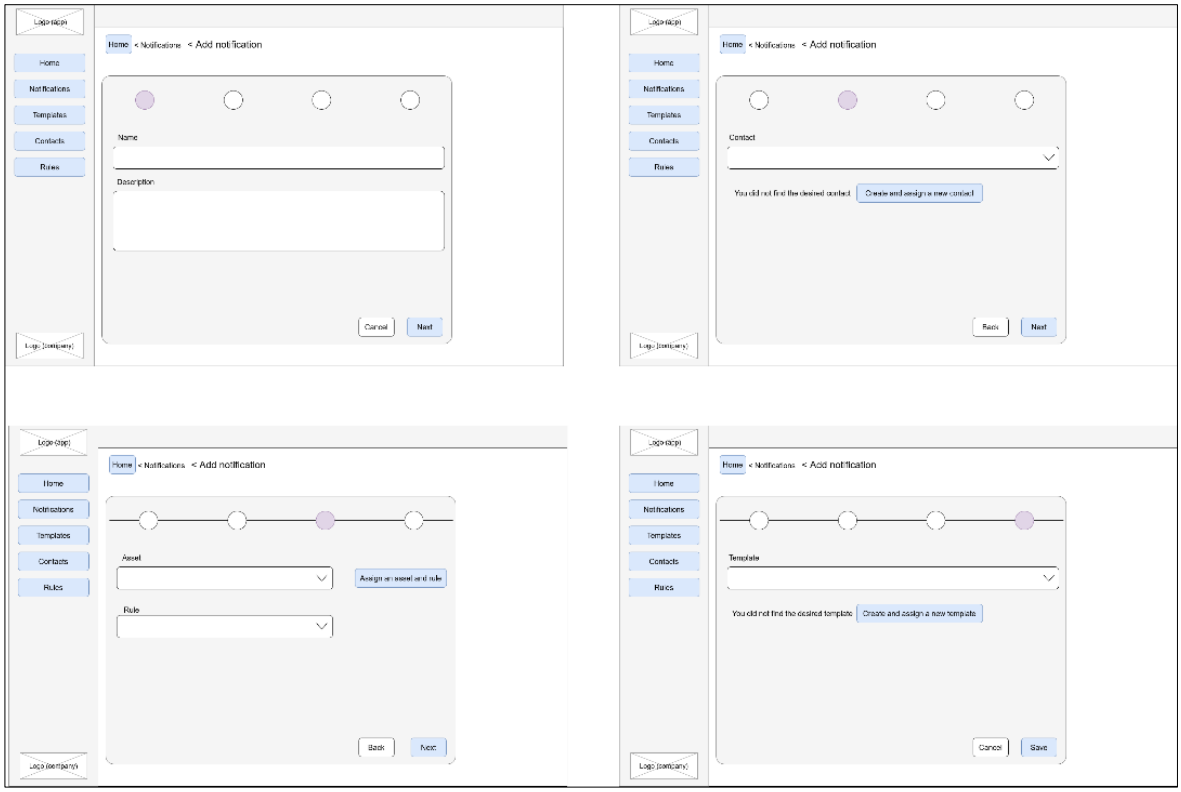
Na přehledu notifikací se zobrazí všechny notifikace, které jsou uložené v databázi. Uživateli je poskytnut vyhledávač pro rychle vyhledání notifikací dle jeho preferencí. Po kliknutí uživatele na vybranou notifikaci se v pravé části obrazovky zobrazí karta detailu dané notifikace, kde jsou zobrazeny všechny parametry. Dále má uživatel možnost na kartě detailu notifikace upravit danou notifikaci, nebo ji smazat pomocí příslušných tlačítek na kartě. Na této stránce má uživatel také možnost přejít na stránku při přidání notifikace.



Obrázek č. 18 – Wireframe přehledu notifikací

4.5.3 Přidání notifikace

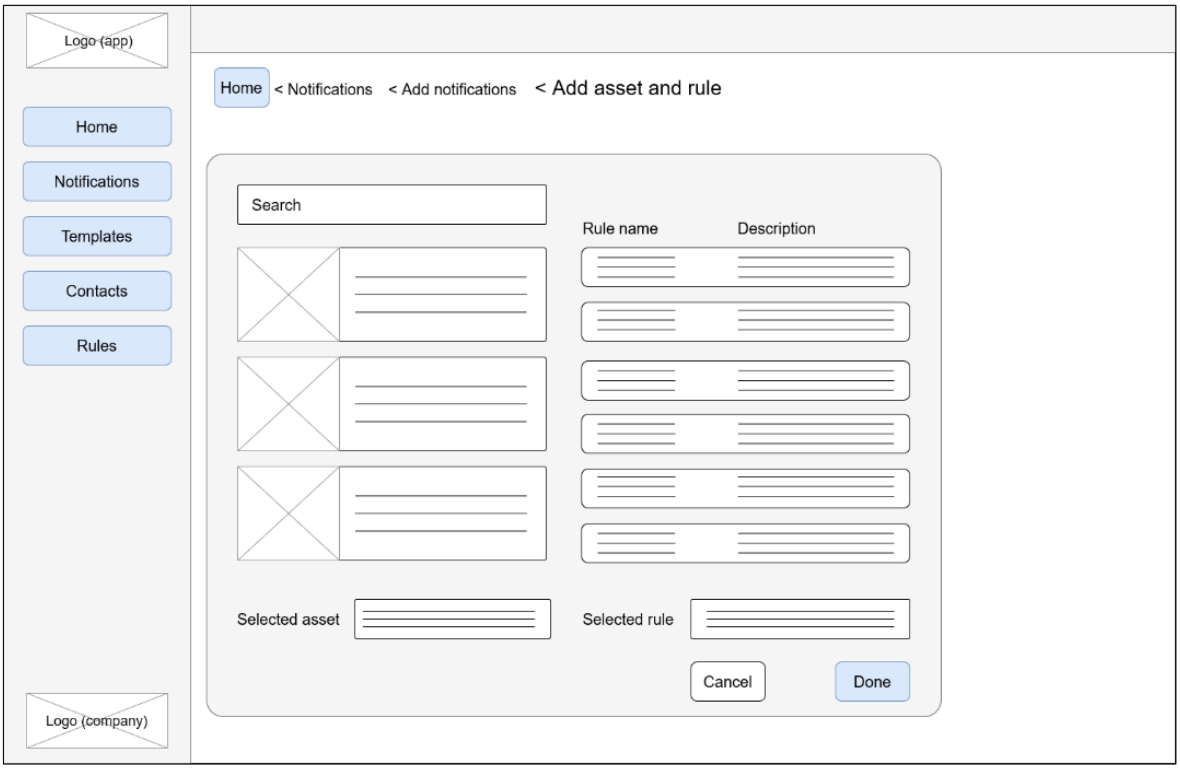
Sekce pro přidání notifikace je rozdělena do čtyř částí. V první části uživatel přidává základní informace o notifikaci, jako je název a krátký popis. V druhé části uživatel vybírá kontakty, které má aplikace přidané v databázi. V třetí části uživatel vybírá asset a požadované pravidlo k danému assetu. V poslední části přidání si uživatel zvolí e-mailovou šablonu, která se zašle na zvolený kontakt v případě nastání události zvoleného pravidla.



Obrázek č. 19 – Wireframe přidání notifikace

4.5.4 Přidání notifikace – zvolení assetu a pravidla

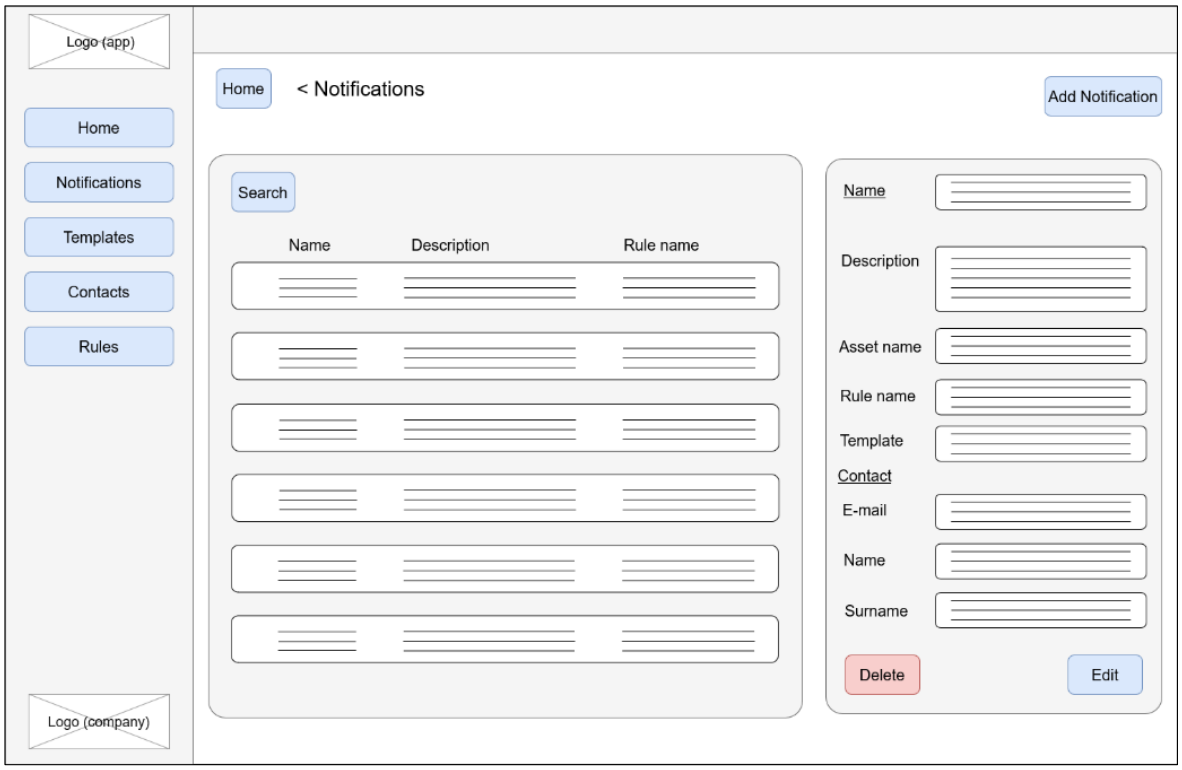
V této části přidání notifikace je uživatelem zobrazen výpis assetů, které jsou uloženy v OS MindSphere. Uživateli je poskytnut vyhledávač pro rychlé vyhledání assetu dle jeho preferencí. Po kliknutí uživatele na vybraný asset se v pravé části karty zobrazí všechna vytvořená pravidla daného assetu, které uživatel může vybrat pro tvorbu notifikace.



Obrázek č. 20 – Wireframe přidání notifikace – zvolení assetu a pravidla

4.5.5 Přehled kontaktů

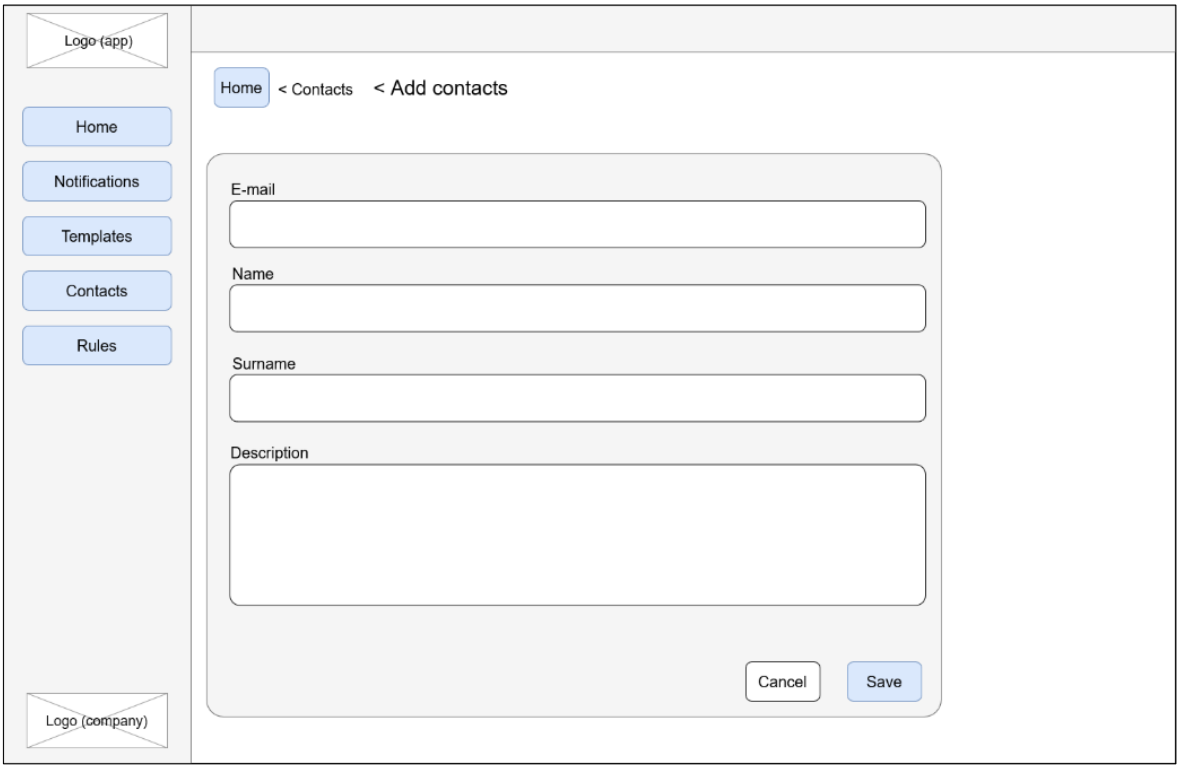
Na přehledu kontaktů se zobrazí všechny kontakty, které jsou uložené v databázi. Uživateli je poskytnut vyhledávač pro rychlé vyhledání kontaktu dle jeho preferencí. Po kliknutí uživatele na vybraný kontakt se v pravé části obrazovky zobrazí detailní karta, na které jsou zobrazeny všechny parametry. Pokud má uživatel oprávnění aplikace typu administrátor, pak má možnost na kartě detailu kontaktu upravit nebo smazat daný kontakt pomocí příslušných tlačítek. Na této stránce má uživatel typu administrátor také možnost přejít na stránku pro přidání kontaktu.



Obrázek č. 21 – Wireframe přehledu notifikací

4.5.6 Přidání kontaktu

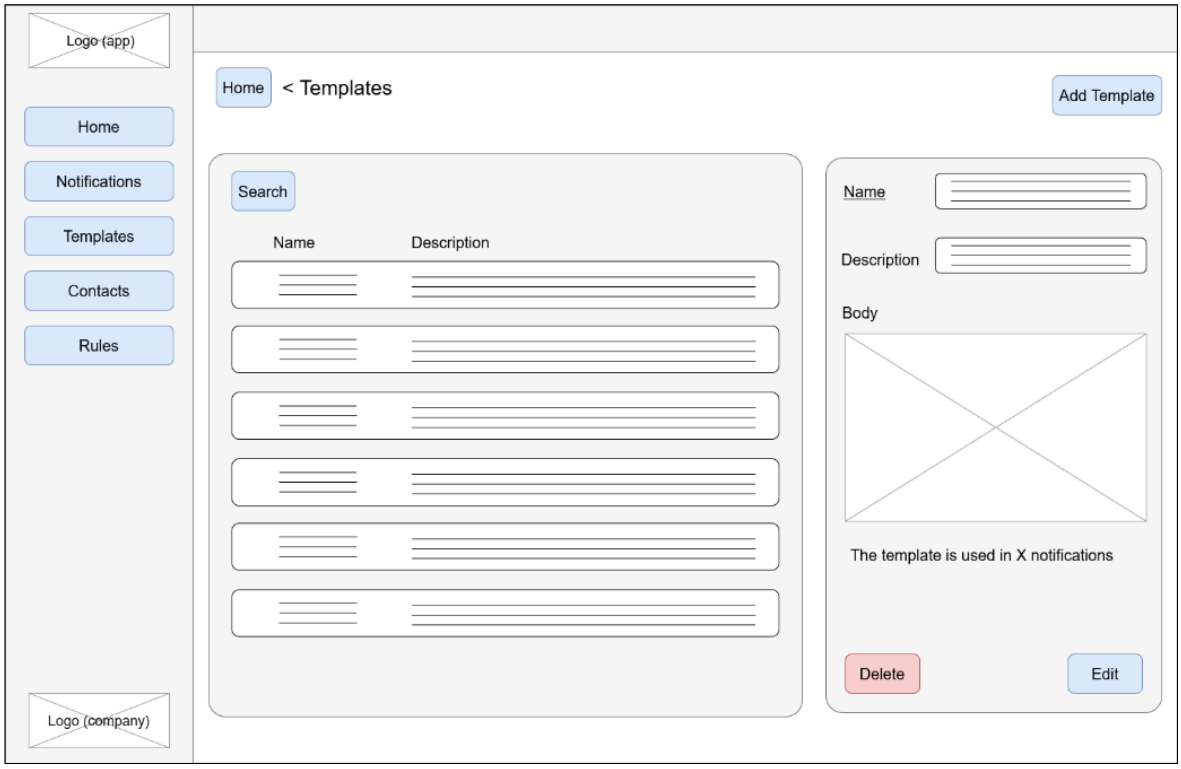
Na stránce přidání kontaktu se zobrazí formulář s parametry, které je potřeba vyplnit pro úspěšné přidání kontaktu. Jedná se o vyplnění e-mailové adresy, jména, příjmení a popisu kontaktu.



Obrázek č. 22 – Wireframe přidání kontaktu

4.5.7 Přehled šablon

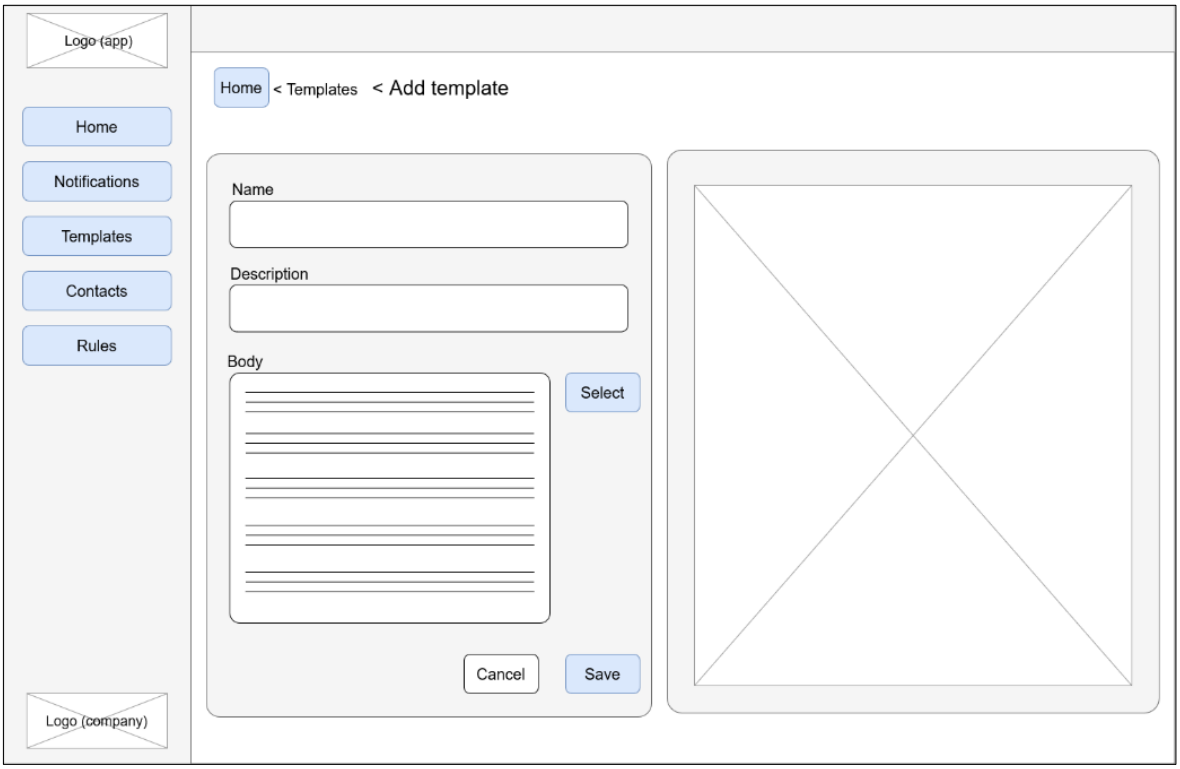
Na přehledu šablon se zobrazí všechny šablony, které jsou uložené v databázi. Uživateli je poskytnut vyhledávač pro rychlé vyhledání šablony dle jeho preferencí. Po kliknutí uživatele na vybranou šablonu se v pravé části obrazovky zobrazí karta detailu dané šablony, kde jsou zobrazeny všechny parametry. Dále má uživatel možnost na kartě detailu šablony upravit danou šablonu nebo ji smazat pomocí příslušných tlačítek na kartě. Na této stránce má uživatel také možnost přejít na stránku pro přidání nové šablony.



Obrázek č. 23 – Wireframe přehledu šablon

4.5.8 Přidání šablony

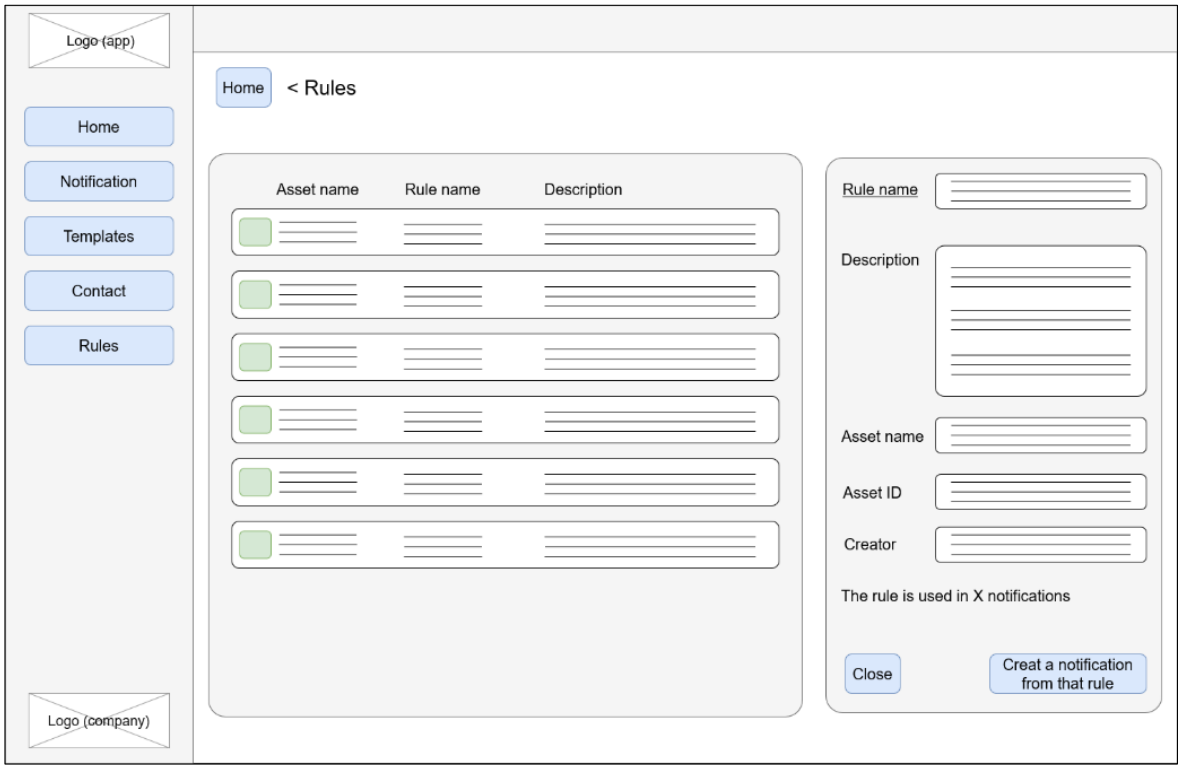
Na stránce přidání šablony se zobrazí formulář s parametry, které je potřeba vyplnit pro úspěšné přidání šablony. Jedná se o vyplnění názvu šablony, popisu a HTML obsahu šablony. Po jakémkoliv zadání obsahu šablony se automaticky vykresluje obsah šablony na kartě v pravé části obrazovky. Uživatel má také možnost nahrát obsah šablony z již připraveného HTML souboru ze zařízení.



Obrázek č. 24 – Wireframe přidání kontaktu

4.5.9 Přehled pravidel

Na přehledu pravidel se zobrazí všechna pravidla, která jsou uložena v MindSphere. Uživateli je poskytnut vyhledávač pro rychlé vyhledání pravidla dle jeho preferencí. Po kliknutí uživatele na vybrané pravidlo se v pravé části obrazovky zobrazí karta detailu daného pravidla, kde jsou zobrazeny všechny parametry. Dále má uživatel možnost na kartě detailu pravidla možnost přejít na tvorbu notifikace s již vybraným pravidlem.



Obrázek č. 25 – Wireframe přehledu pravidel

4.6 Implementace

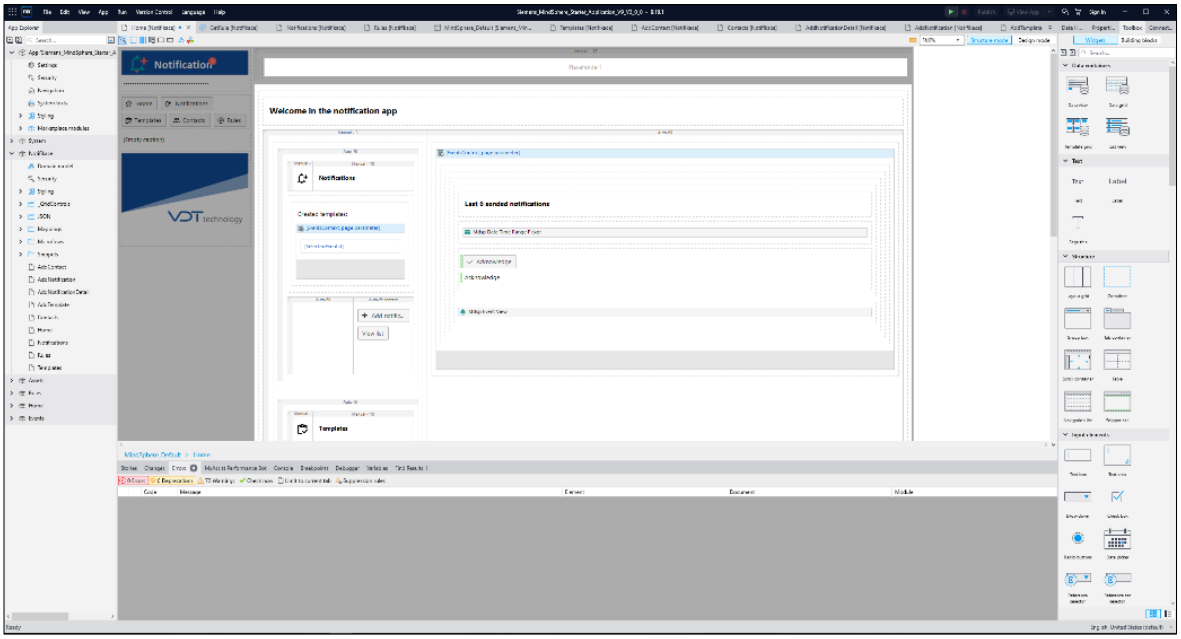
4.6.1 Prvotní nastavení

Nejprve je vytvořen nový projekt v aplikaci Mendix Studio Pro. Následně je stažen v Mendix store základní layout aplikace pro MindSphere. Poté jsou změněny všechny potřebné parametry aplikace jako je název aplikace. Dalším krokem je vygenerování nasazovacího balíčku, který je možné nahrát do open-sourcové, multicloudové aplikace Cloud Foundry. Po přihlášení do Cloud Foundry je vytvořen prostor pro aplikaci. Následně je vyhledán v souborovém systému zařízení nasazovací balíček, který je nahrán do vymezeného prostoru pomocí příkazu: `cf push`. K danému balíčku je potřeba připojit databázi pomocí příkazu: `cf create-service postgresql10 postgresql-xs notifikace-db`. Následně se provede prvotní registrace aplikace v MindSphere a jsou vygenerovány přihlašovací údaje k aplikaci tzv. credentials, díky kterým je možné navázat spojení do prostředí MindSphere i při spuštění aplikace na localhostu.

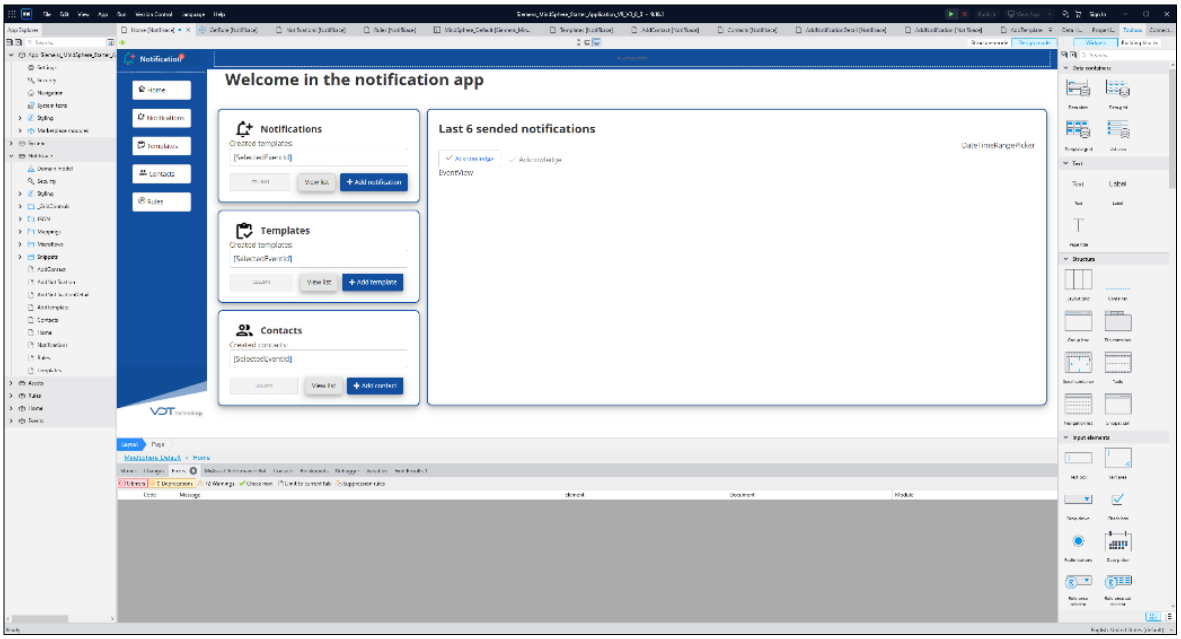
4.6.2 Tvorba stránek

Prvotně je vyvinuta základní kostra všech stránek dle drátěného modelu a je vytvořen design aplikace. Dalším krokem je vytvoření potřebných datových struktur skrze doménový model, díky kterým je možné pracovat s daty ve vytvářených funkcích skrze Microflows.

Jako první je vytvořena úvodní strana se všemi požadovanými funkcemi. U každé komponenty na dané stránce je zvoleno oprávnění. Tímto způsobem jsou vyvíjeny všechny ostatní stránky se všemi komponentami příslušných stránek. Komponenty jsou na stránkách provázány s Microflows nebo přímo s konkrétními entitami doménového modelu.

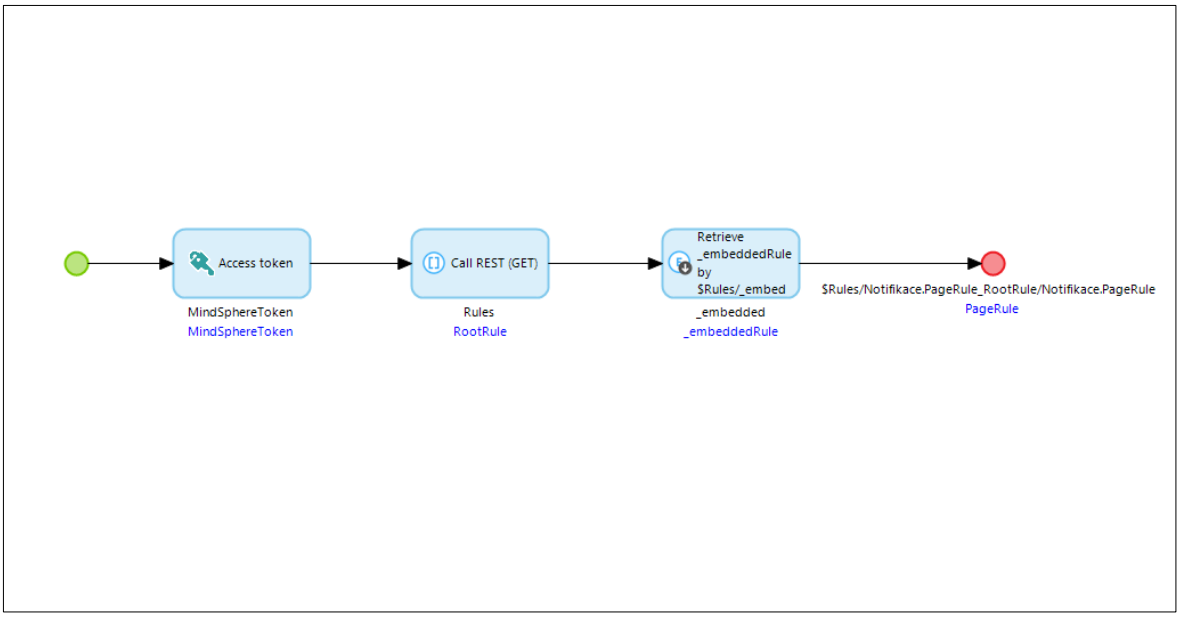


Obrázek č. 26 – Mendix Studio Pro – Structure mode úvodní obrazovky



Obrázek č. 27 – Mendix Studio Pro – Design mode úvodní obrazovky

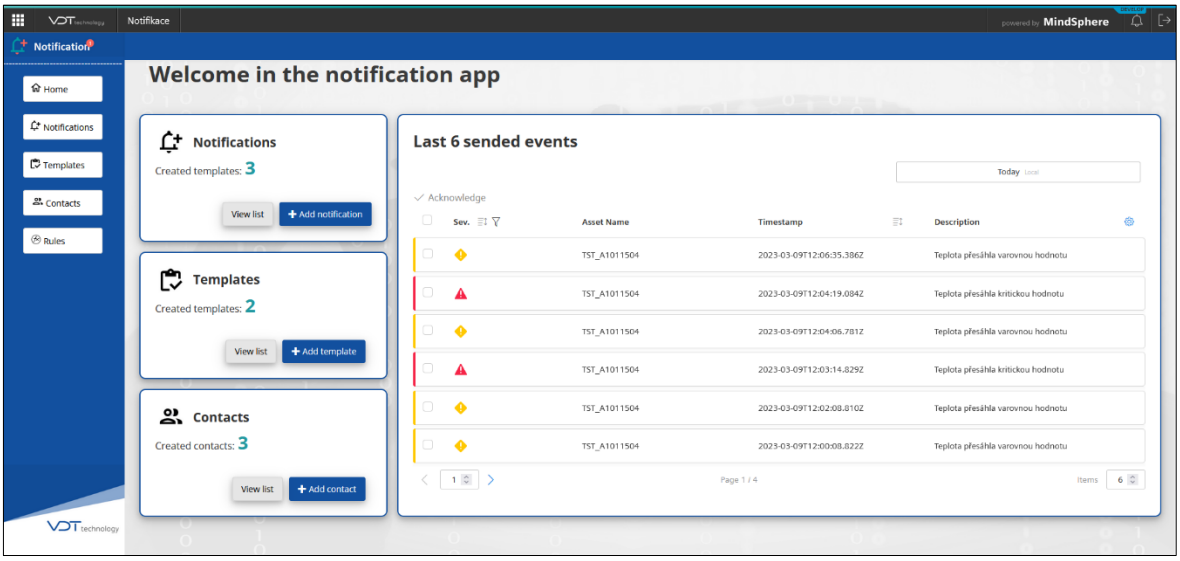
Na obrázku číslo 28 je možné vidět ukázkové Microflows. Tato Microflows zajišťují volání požadavku do MindSphere. REST API požadavek využívá metodu GET, která vrací všechna pravidla, která jsou v MindSphere uložena. Odpověď požadavku je datového typu JSON, který je následně vložen do neperzistentní struktury v Mendixu, kde je dále využíván.



Obrázek č. 28 – Mendix Studio Pro – Ukázkové Microflow

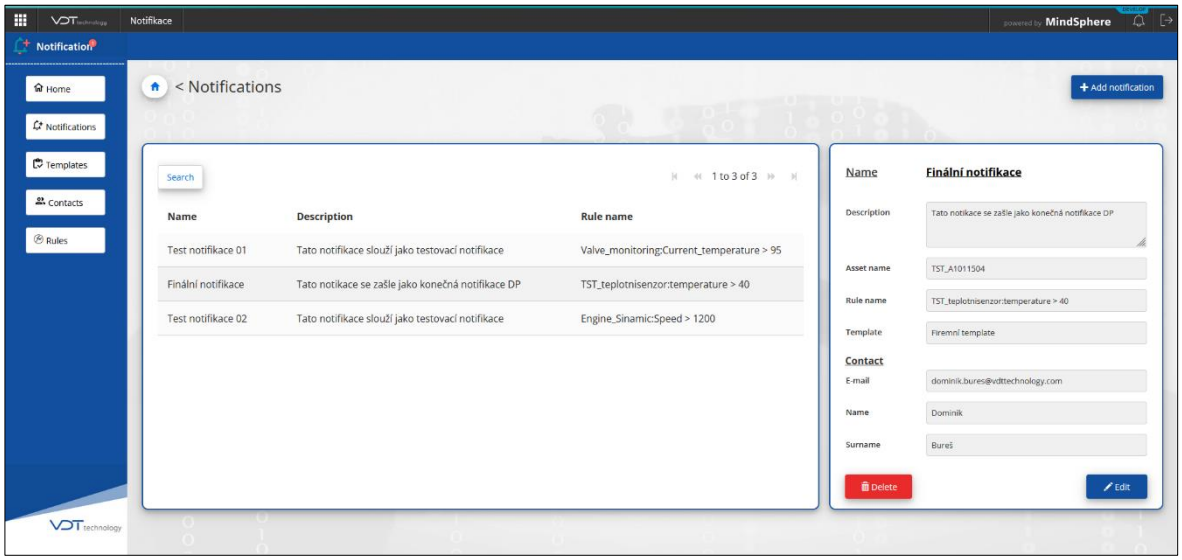
4.6.3 Finální vzhled aplikace

Na obrázku číslo 29 je možné vidět finální vzhled úvodní obrazovky. Hlavní částí obrazovky jsou události, které nastaly a které jsou rozlišené podle jejich závažnosti.



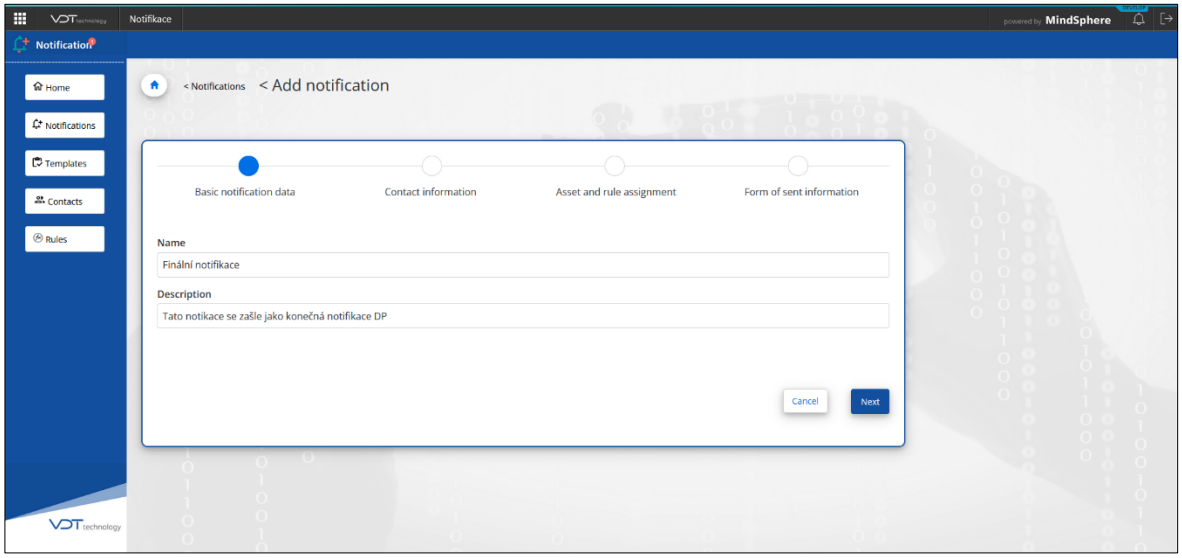
Obrázek č. 29 – Finální úvodní strana

Obrázek číslo 30 představuje finální přehled notifikací. Po vybrání notifikace se zobrazí její detail, který obsahuje jak informace o samotné notifikaci, jako jsou jméno a popis, tak i informace o vybraném assetu, pravidlu, šablony i kontaktu.



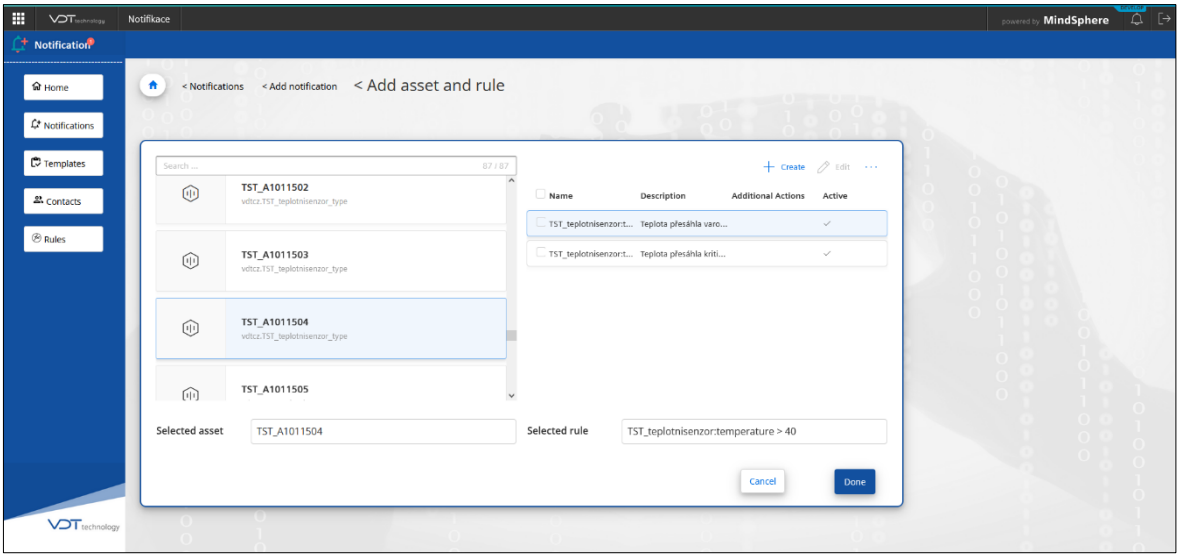
Obrázek č. 30 – Finální přehled notifikací

První ze čtyř částí, které je nutné vyplnit pro přidání notifikace je možné vidět na obrázku číslo 31. V této části je potřeba vyplnit základní informace o notifikaci jako je název a popis notifikace.



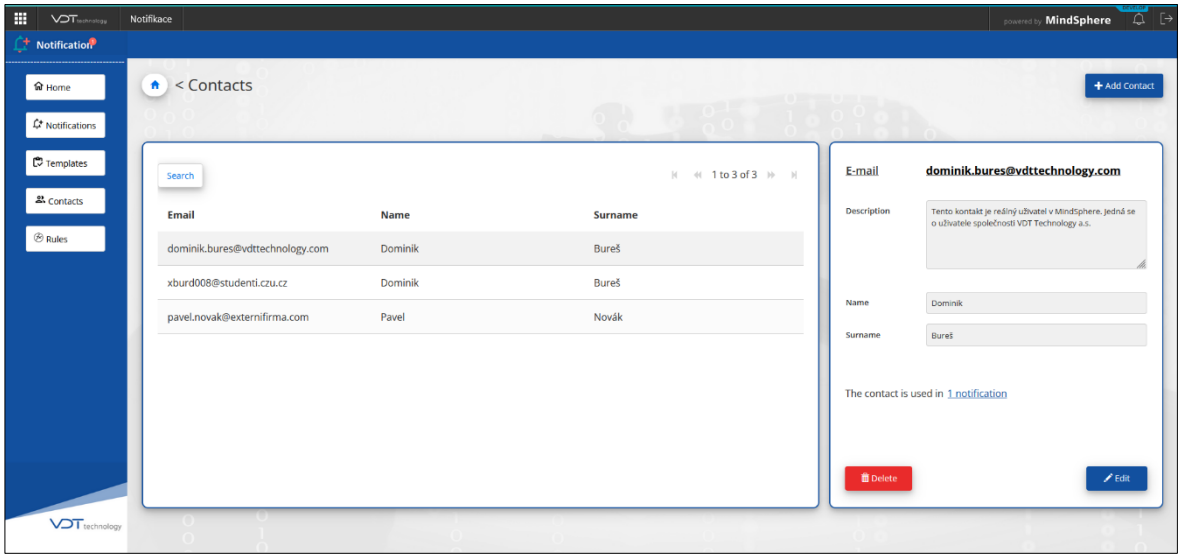
Obrázek č. 31 – Finální přidání notifikace

Pro přidání assetu a pravidla, které jsou potřebné pro vytvoření notifikace jsou tyto informace načítány přímo z MindSphere. Nejprve je zaslán API požadavek na všechny assety v MindSphere. Po vybrání assetu uživatelem je zaslán z aplikace požadavek na všechna pravidla daného assetu v MindSphere. Finální vizualizaci je možné vidět na obrázku číslo 32.

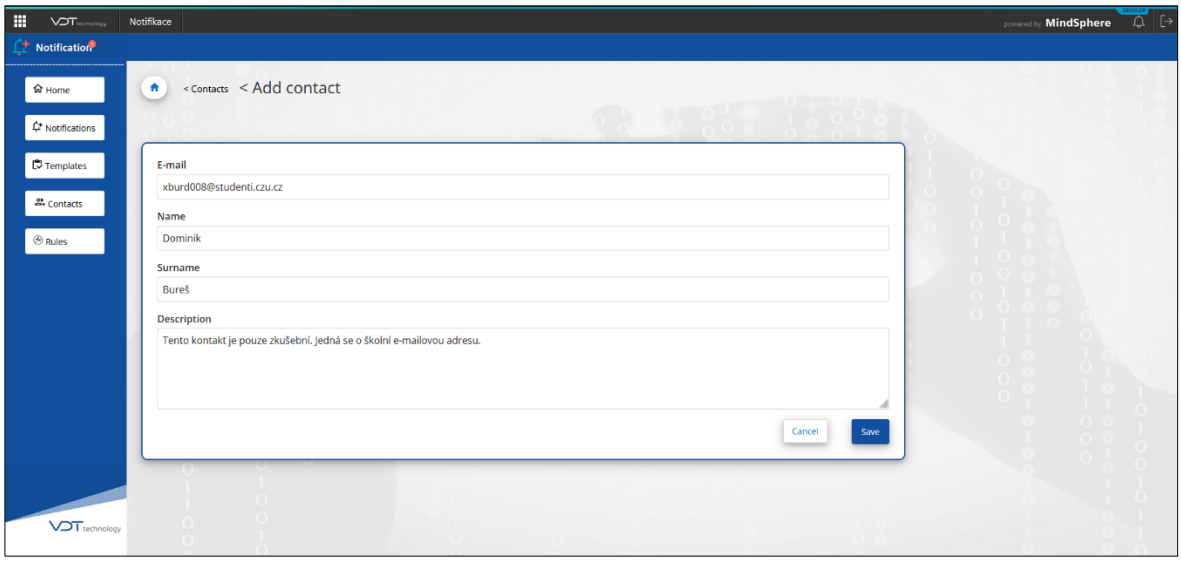


Obrázek č. 32 – Finální přidání notifikace – detail

Finální vizualizaci přehledu všech kontaktů je možné vidět na obrázku číslo 33. Po vybrání konkrétního kontaktu uživatelem je zobrazen detail kontaktu, kde jsou informace o kontaktu jako jsou e-mailová adresa, na kterou se bude notifikace zasílat (v případě vybrání kontaktu při vytváření notifikace), jméno a příjmení a také krátký popis kontaktu. Finální obrazovku pro přidání kontaktu je možné vidět na obrázku číslo 34. Možnost přidání kontaktu je vyhrazeno pouze s oprávněním typu administrátor, dle požadavků na aplikaci.

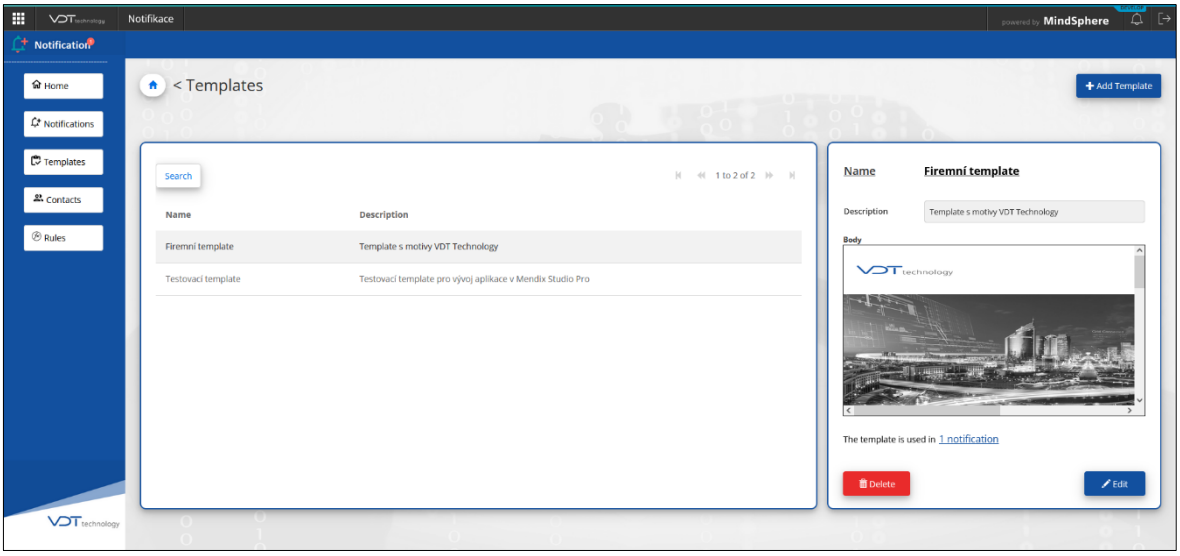


Obrázek č. 33 – Finální přehled kontaktů



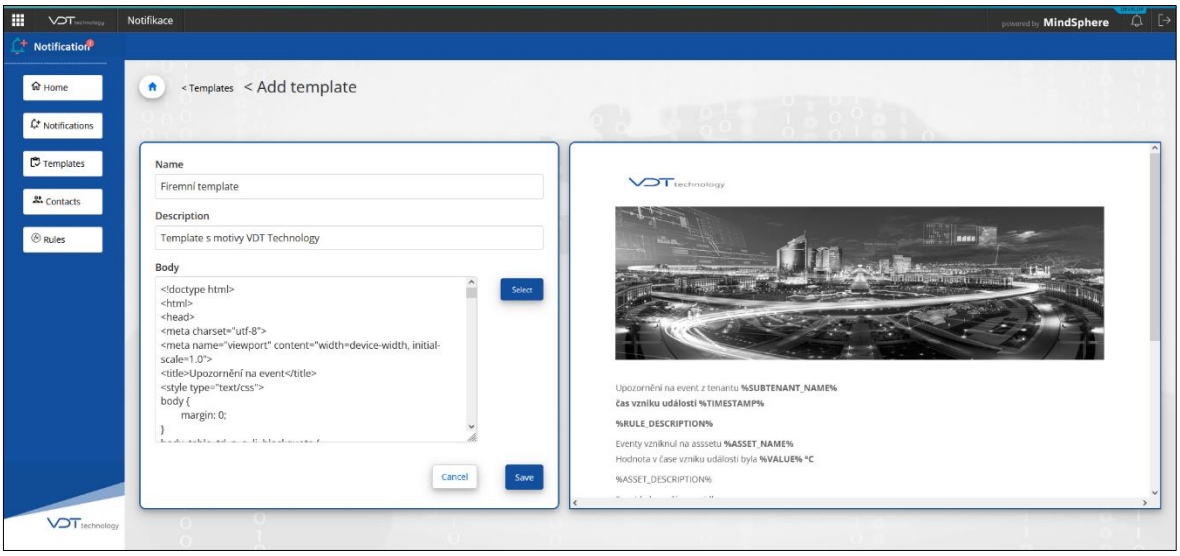
Obrázek č. 34 – Finální přidání kontaktu

Přehled šablon je vyobrazen na obrázku číslo 35. Vizualizace a rozložení stránky je na stejném principu jako jsou kontakty, notifikace či pravidla. Po vybrání konkrétní šablony je zobrazen detail šablony v pravé části obrazovky. Je zobrazen název šablony, popis a interpretované HTML jako tělo šablony. Také je na kartě detailu informace o tom, v kolika vytvořených notifikacích je daná šablona použita.



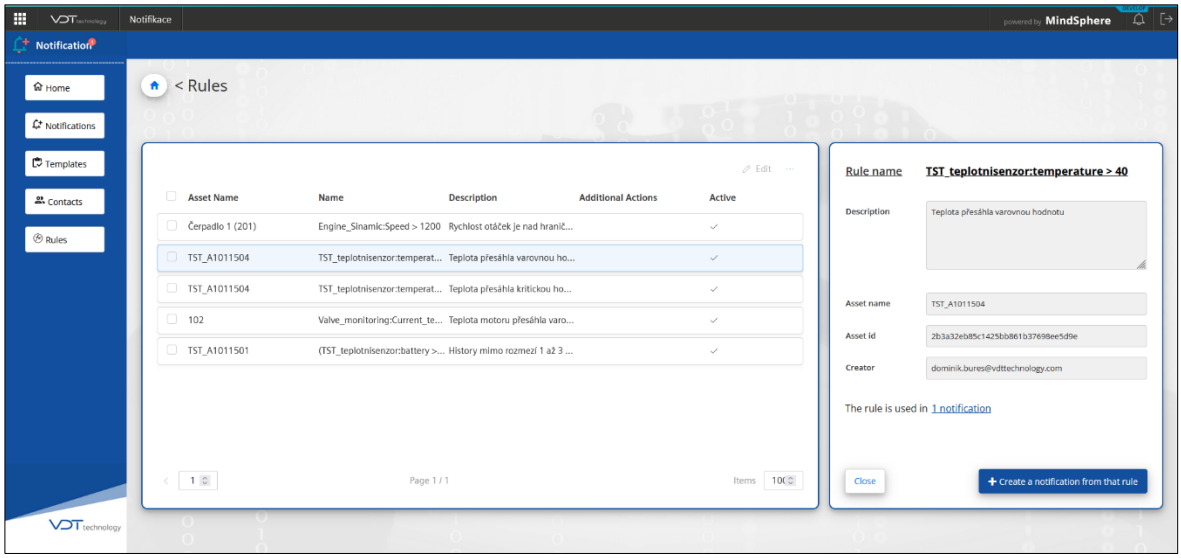
Obrázek č. 35 – Finální přehled šablon

Na obrázku číslo 36 je finální obrazovka přidání šablony. V levé části obrazovky jsou přidávány parametry o šabloně. Uživatel má možnost importovat připravený HTML soubor přímo ze zařízení nebo psát HTML přímo do textové oblasti pro tělo šablony. Přidaná či psaná šablona se ihned vizualizuje v pravé části obrazovky, takže uživatel má možnost ihned vidět, jak bude daná šablona vypadat v zaslaném mailu.



Obrázek č. 36 – Finální přidání šablony

Přehled pravidel je oproti šablonám, kontaktům či notifikacím na jiném principu. Pravidla jsou volána z aplikace prostřednictvím API do MindSphere. Oproti tomu šablony, kontakty a notifikace jsou ukládány přímo v databázi aplikace. Po vybrání daného pravidla je zobrazen detail, který obsahuje název pravidla, popis, název assetu, ke kterému je pravidlo vytvořeno, ID assetu a tvůrce daného pravidla. Finální obrazovku přehledu pravidel je možné vidět na obrázku číslo 37.



Obrázek č. 37 – Finální přehled pravidel

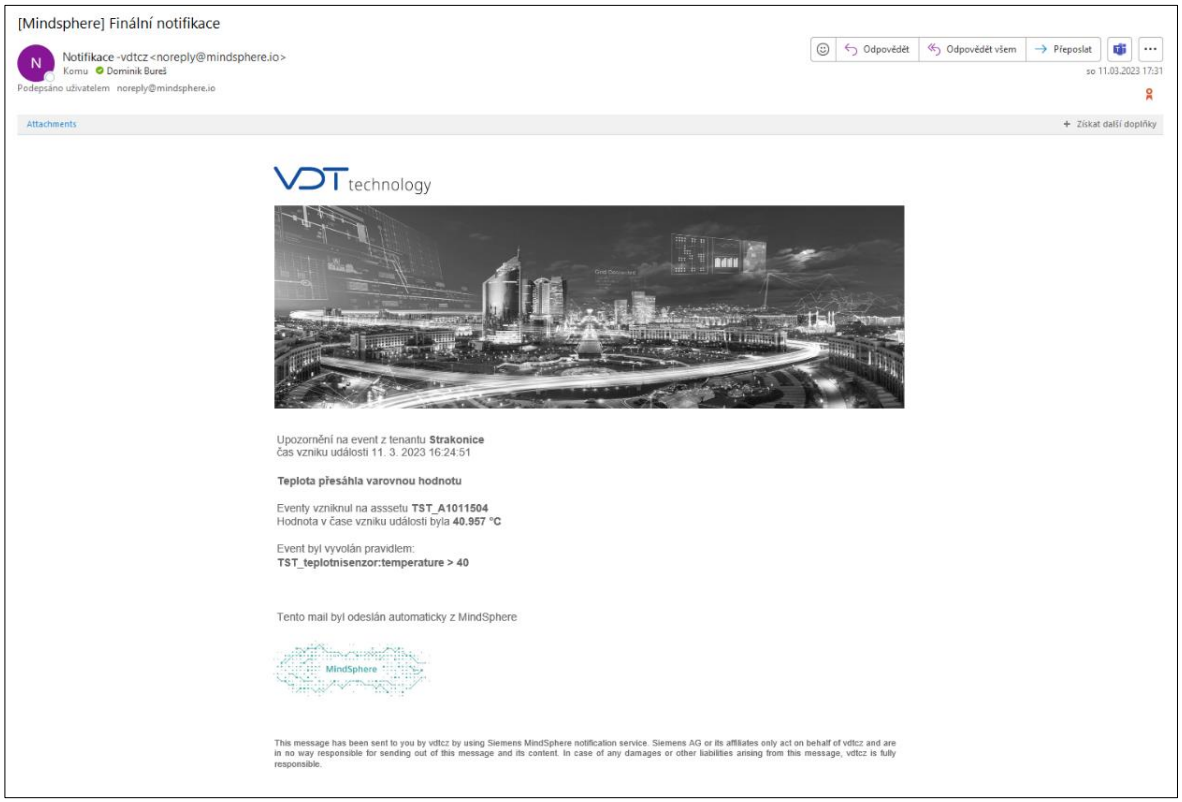
4.7 Testování

Prvotní testování je provedeno již při samotném vývoji aplikace. Při každém dokončení požadované funkce je provedeno několik testů, zda daná funkce pracuje správně a dle požadavků. Tímto způsobem jsou provedeny testy všech funkcí v rámci celé aplikace.

Hlavní testování je realizováno po dokončení všech stránek aplikace se všemi komponentami, které jsou vzájemně provázány s Microflows nebo přímo s entitami doménového modelu.

Další testování je prováděno tak, že je vytvořeno pravidlo na zvoleném libovolném assetu v prostředí MindSphere. Následně je v aplikaci vytvořena notifikace se zvoleným assetem a pravidlem. Je nastaven kontaktní e-mail, kam se má daná notifikace zasílat. Dále je přidána šablona, která bude automaticky naplněna při nastání události v MindSphere. Následně jsou uměle generována data tak, aby došlo k aktivování daného pravidla, díky kterému je vytvořena událost. Tato událost je zaznamenána aplikací, která si vyhledá vytvořenou notifikaci, vloží hodnoty parametrů do připravené šablony, zašle notifikaci na danou e-mailovou adresu a následně zašle požadavek do MindSphere na změnu parametru události acknowledge na hodnotu true. Po zkontrolování nastavení parametru události na true a příchozím mailu je test téměř dokončen. V poslední části provedeného testu je provedena kontrola, zda zaslaný mail obsahuje správně zaznamenané hodnoty události v MindSphere. Následně je toto testování provedeno ještě devětkrát, aby bylo lépe ověřeno, že aplikace prokazuje požadované chování.

4.7.1 Uživatelské testování



Obrázek č. 38 – Vzhled notifikace zaslané na e-mail

4.7.2 Nasazení aplikace do OS MindSphere

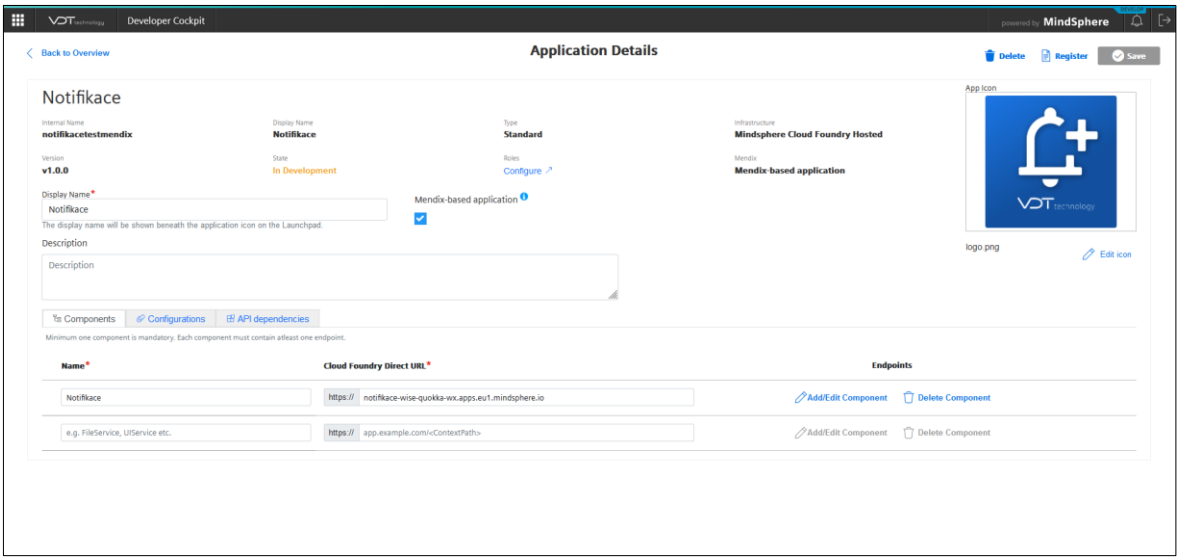
Nasazení aplikace do OS MindSphere je uskutečněno prostřednictvím Cloud Foundry. Nejprve je provedeno přihlášení do Cloud Foudry. Následně je vyhledán prostor, kam se prvotně nahrála aplikace. Následně je vyhledán v souborovém systému zařízení soubor obsahující vyvinutou aplikaci. Pomocí příkazu: `cf push` je spuštěn proces nasazování. Po úspěšném nasazení je možné vidět, že aplikace je spuštěna a je možné přejít k dalšímu kroku.

```
API endpoint:   https://api.cf.eu1.mindsphere.io (API version: 2.195.0)
User:           dominik.bures@vdttechnology.com
Org:            vdtcz
Space:          notifikace

C:\Users\Doo>cf apps
Getting apps in org vdtcz / space notifikace as dominik.bures@vdttechnology.com...
OK

name         requested state   instances   memory   disk   urls
notifikace   started           1/1         256M     256M   notifikace-wise-quokka-wx.apps.eu1.mindsphere.io
```

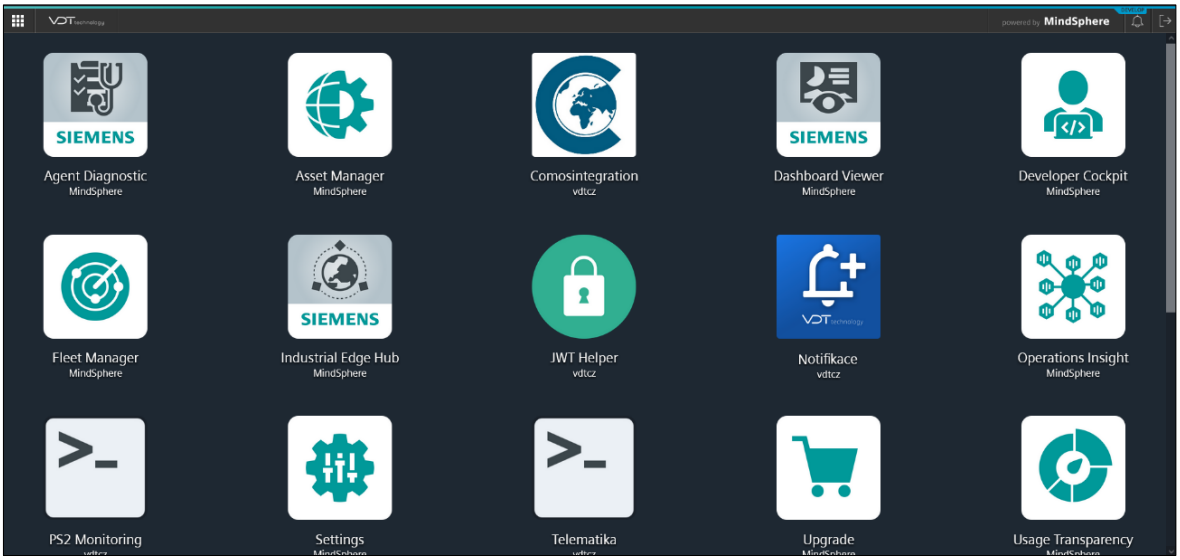
Obrázek č. 39 – Aplikace v Cloud Foundry

Po úspěšném nasazení aplikace do Cloud Foundry je potřeba aplikaci zaregistrovat v tzv. Developer Cockpitu přímo v MindSphere. Zde je potřeba vyplnit jméno aplikace,

zaškrtnout, že je aplikace vytvořena pomocí Mendixu a přidat Cloud Foundry URL, kterou je možné vidět na obrázku číslo 39. Následně je potřeba aplikaci zaregistrovat a přidat ji příslušná oprávnění v nastavení MindSphere tak, aby byla viditelná pro konkrétní uživatele.



Obrázek č. 40 – Developer Cockpit – registrace aplikace



Obrázek č. 41 – MindSphere launchpad – aplikace Notifikace

5 Výsledky a diskuse

5.1 Výsledek práce

Výsledkem práce je notifikační aplikace v OS MindSphere, která umožňuje zasílat vlastní vzhled notifikací na zvolené e-mailové adresy. Aplikace byla vyvinuta v prostředí Mendix Studio Pro, kde byla řádně otestována a následně nasazena do OS MindSphere, přesněji do prostoru vymezeného pro společnost VDT Technology a. s. Aplikace splňuje všechny požadavky, které byly vytyčeny při stanovení cílů aplikace. Vyvinutá a nasazená aplikace tak umožňuje uživatelům vytvářet vlastní HTML šablony pro jakékoliv typy událostí, které se generují z pravidel při jejich aktivaci v MindSphere.

5.2 Možnosti zlepšení

V každé vyvinuté aplikaci je možné nalézt potenciál pro další zlepšení. Tuto aplikaci je možné zlepšit v mnoha ohledech. Například u kontaktů je aplikaci možné zlepšit možností přidání více kontaktů ke konkrétní notifikaci při jejím vytváření. Dalším zlepšením by byla možnost přidání SMS kontaktu. To by znamenalo i možnost vytvářet vlastní SMS šablony, které by bylo možné upravovat primárně v rámci zasílaných parametrů v dané SMS šabloně, nikoliv v rámci designu zasílané SMS zprávy.

Dalším zlepšením aplikace by mohla být možnost vybrání více assetů se stejnými pravidly tak, aby uživatel již nemusel přidávat notifikaci vždy pouze k jednomu assetu a jednomu pravidlu. Toho by bylo možné dostáhnout pouze v rámci stejné struktury assetů, tzv. asset typu. Pouze tak by bylo možné zajistit, že všechny vybrané assety obsahují stejné parametry, pro které by bylo možné danou notifikaci nastavit.

Aktuálně není aplikace plně optimalizovaná pro mobilní zařízení, jelikož je převážná většina aplikací v OS MindSphere využívána pouze skrze desktopová zařízení. Avšak bylo by možné tuto aplikaci optimalizovat i pro zobrazení na mobilních zařízeních, a tak rozšířit uživatelům možnosti použití.

6 Závěr

Primárním cílem této diplomové práce bylo provést analýzu uživatelských požadavků na notifikační aplikaci a následně vytvořit návrh vlastní notifikační aplikace, který bude implementován a otestován v OS MindSphere. Vzhledem k tomu, že zákazníci i zaměstnanci VDT Technology požadovali zasílání notifikací v českém jazyce s možností vlastního vzhledu zasílané notifikace, byla již dosud vyvinutá existující řešení nevyhovující, a proto bylo nutné vyvinout řešení vlastní.

Teoretická část byla věnována podrobným představením OS MindSphere a technologiím, které byly následně použity pro vývoj samotné aplikace v praktické části. Dále se teoretická část zabývala analýzou s následným zhodnocením již existujících notifikačních aplikací v OS MindSphere.

Praktická část se zabývala samotným vývojem notifikační aplikace. Nejprve bylo provedeno stanovení cílů aplikace, následně stanovení vrstev oprávnění aplikace. Dále byla provedena analýza uživatelských požadavků na aplikaci, která byla popsána nástroji Use Case. Následně byl vytvořen drátěný model celé aplikace, na jehož základě se začala specifikovaná aplikace vyvíjet. Při vyvíjení aplikace byla aplikace vystavena podrobným testováním a po vyvinutí aplikace byly všechny funkcionality několikrát otestovány. Následně byla aplikace plnohodnotně nasazena do OS MindSphere.

7 Seznam použitých zdrojů

1. Cloud Computing: Co to je a komu se vyplatí | Algotech.cz. IT (ICT) služby na míru - technologie, outsourcing, řešení a správa infrastruktury - Algotech [online]. Copyright ©2019 http [cit. 16.10.2022]. Dostupné z: https://www.algotech.cz/novinky/2020-04-21-cloud-computing-co-to-je-a-komu-se-vyplati

2. Learn Cloud Computing Tutorial - javatpoint. Tutorials List - Javatpoint [online]. Copyright © Copyright 2011 [cit. 16.10.2022]. Dostupné z: https://www.javatpoint.com/cloud-computing-tutorial

3. The Cloud Computing market promises to grow in 2016. In: *Sky.One Solutions - Platforms for the digital evolution of your business* [online]. São Paulo: Amplifica Digital, 2016 [cit. 2023-03-14]. Dostupné z: https://skyone.solutions/en/hub/the-cloud-computing-market-promises-to-grow-in-2016/

4. MÁCHA, Petr. Historie a základní principy cloud computingu. *SystemOnLine.cz - ekonomické a informační systémy v praxi* [online]. Brno: CCB spol, 2015 [cit. 2022-10-16]. Dostupné z: https://www.systemonline.cz/virtualizace/historie-a-zakladni-principy-cloud-computingu.htm

5. Co je cloud computing a jak funguje?. *Najít Nejlepší Projekty...* [online]. West Hollywood: Hundeshagen Digital Media, 2018 [cit. 2022-10-16]. Dostupné z: https://cs.jf-parede.pt/what-is-cloud-computing

6. Tenký klient - ITBiz.cz. Zprávy ze světa IT a byznysu - ITBiz.cz [online]. Copyright © 2019 Vydává [cit. 16.10.2022]. Dostupné z: https://www.itbiz.cz/slovnik/informacni-technologie-it/tenky-klient

7. CHRISTENSSON, Per. Definice tlustého klienta. *Počítačový slovník TechLib* [online]. Sharpened Productions, 2006 [cit. 2022-10-16]. Dostupné z: https://tech-lib.eu/definition/thickclient.html

8. Vícevrstvá architektura: tenký, tlustý a chytrý klient - CleverAndSmart Management Consulting. CleverAndSmart Management Consulting [online]. Copyright © 2008 [cit. 16.10.2022]. Dostupné z: https://www.cleverandsmart.cz/vicevrstva-architektura-tenky-tlusty-a-chytry-klient/

9. Co jsou to datacentra a serverovny | Interval.cz. Interval.cz | Svět Internetu, Technologií a Bezpečnosti [online]. Copyright © [cit. 16.10.2022]. Dostupné z: https://www.interval.cz/clanky/co-jsou-to-datacentra-a-serverovny/

10. What is Distributed Cloud | IBM. *IBM - Country - Czech Republic | IBM* [online]. New York: IBM, 2020 [cit. 2022-10-16]. Dostupné z: https://www.ibm.com/cloud/learn/distributed-cloud

11. Cloud Computing Technology : Basic Functionality, Types & Advanatages. *ElProCus - Electronic Projects for Engineering Students...* [online]. Haidarábád: ElProCus Technologies, c2013-2022 [cit. 2022-10-16]. Dostupné z: https://www.elprocus.com/cloud-computing-the-great-revolution-for-it-industry/

12. FURHT, Borko. *Handbook of Cloud Computing*. 1rd. New York: Springer Science+Business Media, 2010. ISBN 978-1-4419-6523-3.

13. Co je komunitní cloud? - Správa.sítě.eu. Správa sítě - slovník pojmů: správa sítě, zabezpečení sítě, outsourcing IT [online]. Copyright © [cit. 16.10.2022]. Dostupné z: https://www.sprava-site.eu/komunitni-cloud/

14. KAVIS, Michael. *ARCHITECTING THE CLOUD*. 1rd. New Jersey: John Wiley, 2014. ISBN 978-1-118-69177-9.

15. LABERIS, Bill. *What Is the Cloud?*. 2019. Newton: O'Reilly Media, 2019. ISBN 9781492052906.

16. Red Hat. In: *What is IaaS?* [online]. Raleigh: Red Hat, 2022, June 22 [cit. 2023-03-14]. Dostupné z: https://www.redhat.com/de/topics/cloud-computing/what-is-iaas

17. Red Hat. In: *Was ist PaaS?* [online]. Raleigh: Red Hat, 2022, August 26 [cit. 2023-03-14]. Dostupné z: https://www.redhat.com/de/topics/cloud-computing/what-is-paas

18. Red Hat. In: *Was ist SaaS?* [online]. Raleigh: Red Hat, 2022, March 25 [cit. 2023-03-14]. Dostupné z: https://www.redhat.com/de/topics/cloud-computing/what-is-saas

19. What is XaaS (Anything as a Service)?. Purchase Intent Data for Enterprise Tech Sales and Marketing - TechTarget [online]. Dostupné z: https://www.techtarget.com/searchcloudcomputing/definition/XaaS-anything-as-a-service

20. Siemens Česká republika. *Co je MindSphere?* [online]. Mnichov: Siemens, 2023 [cit. 2023-03-14]. Dostupné z: https://new.siemens.com/cz/cs/reseni/mindsphere.html

21. SiePortal. In: *MindSphere (In Cloud)* [online]. Mnichov: Siemens, 2022 [cit. 2023-03-14]. Dostupné z: https://mall.industry.siemens.com/mall/en/nl/Catalog/Products/10348389#

22. MindSphere: vstupte do světa internetu věcí. *SIEMENS Visions | Magazín o lidech, technologiích a inovacích* [online]. Praha: Siemens, 2018 [cit. 2022-10-16]. Dostupné z: https://www.visionsmag.cz/mindsphere-operacni-cloudovy-system-pro-internet-veci

23. Siemens Digital Industries Software, 2022. *Siemens Software* [online]. [cit. 2023-03-08]. Dostupné z: https://www.plm.automation.siemens.com/global/en/

24. MindSphere security model, 2018. In: *Siemens Software* [online]. [cit. 2023-03-08]. Dostupné z: https://assets.new.siemens.com/siemens/assets/api/uuid:6b876b5e-5594-4da4-90e0-e9e0c6f1f1e1/version:1557483304/siemens-plm-mindsphere-security-model-wp-75966-a7.pdf

25. MindSphere security model, 2018. In: *Siemens Software* [online]. [cit. 2023-03-08]. Dostupné z: https://assets.new.siemens.com/siemens/assets/api/uuid:6b876b5e-5594-4da4-90e0-e9e0c6f1f1e1/version:1557483304/siemens-plm-mindsphere-security-model-wp-75966-a7.pdf

26. MindSphere security model, 2018. In: *Siemens Software* [online]. [cit. 2023-03-08]. Dostupné z: https://assets.new.siemens.com/siemens/assets/api/uuid:6b876b5e-5594-4da4-90e0-e9e0c6f1f1e1/version:1557483304/siemens-plm-mindsphere-security-model-wp-75966-a7.pdf

27. MindSphere: Otevřený operační systém a cloudové řešení pro internet věcí, 2018. In: *Siemens Software* [online]. [cit. 2023-03-08]. Dostupné z: https://assets.new.siemens.com/siemens/assets/api/uuid:6fa69a2da4ac6c37c2a224f713871ddd33925940/version:1520420739/brozuramindsphereczk.pdf

28. Siemens crea una red de centros de aplicaciones MindSphere, 2017. In: *InfoPLC* [online]. [cit. 2023-03-08]. Dostupné z: https://www.infoplc.net/plus-plus/empresas/item/105002-siemens-crea-una-red-de-centros-de-aplicaciones-mindsphere

29. MindSphere: The cloud-based, open IoT operating systém for digital transformation, 2017. In: *Siemens Software* [online]. [cit. 2023-03-08]. Dostupné z: https://cdn2.hubspot.net/hubfs/1147371/Resources/Siemens_MindSphere_Whitepaper.pdf

30. MindSphere Application Center. In: *Siemens Software* [online]. [cit. 2023-03-08]. Dostupné z: https://new.siemens.com/global/en/company/topic-areas/smart-infrastructure/mac4ioe.html

31. Internet věcí (IoT): definice, příklady využití, produkty. WEB & MOBILE DEVELOPMENT AGENCY | Rascasone [online]. Copyright © [cit. 17.10.2022]. Dostupné z: https://www.rascasone.com/cs/blog/iot-internet-veci-definice-produkty-historie

32. Giải pháp tự động hóa IoT. In: *Các Project Smart Home đơn giản cho học sinh, sinh viên* [online]. Hà Nội: MC&TT Co, 2022 [cit. 2023-03-14]. Dostupné z: https://giaiphap.mctt.com.vn/smart-home-project-don-gian/

33. Smart Cities World - Latest news and case studies. In: *Smart cities services worth $225bn by 2026* [online]. London: SmartCitiesWorld, 2017, 26 Apr [cit. 2023-03-14]. Dostupné z: https://www.smartcitiesworld.net/news/news/smart-cities-services-worth-225bn-by-2026-1618

34. Co je to API a jaké jsou možnosti jeho využití?. WEB & MOBILE DEVELOPMENT AGENCY | Rascasone [online]. Copyright © [cit. 17.10.2022]. Dostupné z: https://www.rascasone.com/cs/blog/co-je-api

35. PATNI, Sanjay. *Pro RESTful APIs: Design, Build and Integrate with REST, JSON, XML and JAX-RS*. 2017. Berkeley, CA: Apress, 2017. ISBN 978-1-4842-2664-3.

36. KENNEWEG, Bryan, Imran KASAM a Micah MCMULLEN. *Building Low-Code Applications with Mendix*. Birmingham: Packt Publishing, 2021. ISBN 978-1-80020-142-2.

37. What Is Mendix & How Does It Support the App Development Lifecycle?. Low-code Application Development Platform | Mendix [online]. Copyright © Mendix Technology BV 2022. All rights reserved [cit. 17.10.2022]. Dostupné z: https://www.mendix.com/evaluation-guide/what-is-mendix/

38. Build API | Mendix Documentation. In: *Build API* [online]. Oakville: Mendix, 2023, March 7 [cit. 2023-03-14]. Dostupné z: https://docs.mendix.com/apidocs-mxsdk/apidocs/build-api/

39. Low-code Application Development Platform | Mendix. In: *The Mendix Atlas UI Framework* [online]. Oakville: Mendix, 2023 [cit. 2023-03-14]. Dostupné z: https://www.mendix.com/atlas/

40. ROEST, Danny. Low-code Application Development Platform | Mendix. In: *Mendix 8.2: Go with the (Micro)flow* [online]. Oakville: Mendix, 2019, September 26 [cit. 2023-03-14]. Dostupné z: https://www.mendix.com/blog/mendix-8-2-go-with-the-microflow/

41. Cloud Foundry | VIP Trust s.r.o.. *Úvodní strana | VIP Trust s.r.o.* [online]. Copyright © 2023 VIP Trust s.r.o. [cit. 05.03.2023]. Dostupné z: https://viptrust.com/technologie/ostatni/cloud-foundry

42. BUCHALCEVOVÁ, Alena. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1075-7.

43. SIMATIC Notifier: Product Sheet, 2022. In: *Siemens Software* [online]. [cit. 2023-03-08]. Dostupné z: https://siemens.mindsphere.io/content/dam/mindsphere/documentation/pdf/App_SIMATICNotifier_ProductSheet_v2.0.pdf?ste_sid=7b34bfc9abc57e525c35149559f34a79

44. Fleet Manager. *Overview* [online]. Mnichov: Siemens, c1996-2022 [cit. 2022-10-17]. Dostupné z: https://documentation.mindsphere.io/resources/html/fleet-manager/en-US/107852429835.html
45. REFSNES, Hege. *Learn HTML and CSS with w3schools*. Hoboken: Wiley, c2010. ISBN 978-0-470-61195-1.

46. TERBEROVÁ, Hana. Informace z online světa digitální tvorby, webů a grafiky | Artster. *Co je to CSS – kaskádové styly* [online]. Lipník nad Bečvou: Artster, 2023, 4.3. [cit. 2023-03-25]. Dostupné z: https://artster.cz/

47. BUCHALCEVOVÁ, Alena. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1075-7.

48. ŠTIGLER, Jaroslav. UNIFER - DESIGN & DIGITAL AGENCY. *Drátěný model webu – na rozložení záleží* [online]. Brno: Unifer, 2022, 26. 05. [cit. 2023-03-25]. Dostupné z: https://unifer.cz/

8 Seznam obrázků, tabulek, grafů a zkratek

8.1 Seznam obrázků

8.2 Seznam tabulek