



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**INFORMAČNÍ SYSTÉM PRO ZPRÁVU A ŘÍZENÍ
PROCESŮ WORKFLOW**

INFORMATION SYSTEM FOR WORKFLOW PROCESS MANAGEMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DOMINIK AUGUSTÍN

VEDOUČÍ PRÁCE

SUPERVISOR

Ing. PETR JOHN

BRNO 2024

Zadání bakalářské práce



154665

Ústav: Ústav informačních systémů (UIFS)
Student: **Augustín Dominik**
Program: Informační technologie
Název: **Informační systém pro zprávu a řízení procesů workflow**
Kategorie: Informační systémy
Akademický rok: 2023/24

Zadání:

1. Prostudujte oblast řízení procesů workflow a prozkoumejte stávající metody definice procesů a způsoby jejich zápisu (např. BPMN).
2. Prostudujte problematiku vytváření informačních systémů, způsoby komunikace aplikací pomocí aplikačních rozhraní, typy těchto rozhraní. Prozkoumejte existující nástroje pro řízení procesů (např. Camunda, OpenProject).
3. Analyzujte současný stav a požadavky firmy Ekom pro zefektivnění práce s procesy, způsoby jejich reprezentace a sledovatelnost na moderních technologiích.
4. Dle výsledků analýzy z předchozího bodu, zvolte optimální metody a knihovny a navrhnete řešení pro řízení procesů workflow. Při návrhu aplikace se zaměřte na splnění požadavků firmy Ekom.
5. Navržené řešení implementujte.
6. Proveďte testování funkčnosti a použitelnosti v rámci firmy Ekom.

Literatura:

- Georgakopoulos, Diimitrios., Hornick, Mark. & Sheth, Amit. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distrib Parallel Databases* 3, 119–153 (1995). <https://doi.org/10.1007/BF01277643>
- Aalst, Wil & Ter, Arthur & Kiepuszewski, Bartosz & Barros, Alistair. (2000). *Workflow Patterns*. <https://doi.org/10.1023/A:1022883727209>
- Weske, M. *Business Process Management: Concepts, Languages, Architectures*. Berlin, Heidelberg: Springer Nature, 2007. ISBN 9783540735212. 20
- Introduction | Vue.js [online]. 2023 [cit. 2023-09-10]. Dostupné z: <https://vuejs.org/guide/introduction.html>
- Interní dokumenty firmy Ekom.

Při obhajobě semestrální části projektu je požadováno:
Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **John Petr, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 30.10.2023

Abstrakt

Cielom tejto práce je navrhnuť a implementovať informačný systém pre konkrétnu firmu, kde treba nahradiť stávajúce riešenie. Tento informačný systém sa zameriava na vykonávanie podnikových procesov, inak nazvané procesy workflow, ktoré sú definované v štandarde BPMN. Nový systém sa delí na klientskú časť v podobe webovej aplikácie a na serverovú časť bežiacu v prostredí Node.js. Implementačným jazykom je v oboch prípadoch jazyk TypeScript. Komunikácia medzi súčasťami, ako aj s ďalšími systémami vo firme prebieha rozhraniami štýlu REST. Nový informačný systém firme prináša rozviazanie pevnej a ťažkopádnej podoby vykonávaných procesov, ktoré nahrádza vizuálne konfigurovateľnými diagramami na platforme Camunda.

Abstract

The aim of this thesis is to design and implement an information system for a specific company where the existing solution needs to be replaced. This information system focuses on the execution of business processes, otherwise known as workflows that are defined in the BPMN standard. The new system is divided into a client part in the form of a web application and a server part running in the Node.js runtime environment. The implementation language of both parts is TypeScript. Communication between the system parts as well as with other systems used in the company is done through REST API. The new information system allows the company to leave the rigid and cumbersome form of the executed processes behind, replacing them with visually configurable diagrams run by the Camunda platform.

Klíčové slová

Workflow, BPMN, Typescript, MongoDB, Node.js, REST, Vue

Keywords

Workflow, BPMN, Typescript, MongoDB, Node.js, REST, Vue

Citácia

AUGUSTÍN, Dominik. *Informační systém pro zprávu a řízení procesů workflow*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Petr John

Informační systém pro zprávu a řízení procesů workflow

Prehlásenie

Vyhlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Petra Johna. Ďalšie informácie mi poskytol Ing. Matúš Ivan, správca IT procesov vo firme EKOM spol. s r.o.. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Dominik Augustín

5. mája 2024

Podakovanie

Chcem poďakovať Ing. Petrovi Johnovi za odborné vedenie tejto práce, Ing. Matúšovi Ivanovi za možnosť aplikovania nadobudnutých znalostí v praxi a v neposlednom rade rodine za neustálu a neoceniteľnú podporu.

Obsah

1	Úvod	2
2	Procesy workflow, spôsoby ich definície a zápisu	3
2.1	Moderný podnikový proces	4
2.2	Typy podnikových procesov	5
2.3	Návrh podnikového procesu	5
2.4	Riadenie procesov workflow	6
2.5	Spôsoby zápisu modelov podnikových procesov	7
3	Informačné systémy a nástroje pre riadenie procesov	13
3.1	Architektúry informačných systémov	16
3.2	Spracovanie a vykresľovanie webových stránok	18
3.3	Spôsoby komunikácie aplikácií	20
3.4	Existujúce nástroje pre podporu procesného riadenia	24
4	Analýza súčasného stavu a požiadaviek firmy EKOM	27
4.1	Názory používateľov systému – dotazník	38
5	Návrh nového systému	42
6	Implementácia navrhnutého systému	45
6.1	Databázová vrstva	45
6.2	Aplikačná vrstva – server	47
6.3	Používateľské rozhranie	50
6.4	Nasadenie systému	56
7	Testovanie funkčnosti a použiteľnosti nového systému	57
8	Záver	59
	Literatúra	60
A	Snímky obrazovky aktuálnej verzie systému Workflow vo firme EKOM	63
B	Výsledky dotazníku k aktuálnej verzii systému Workflow pre zamestnancov firmy EKOM	65
C	Workflow 3 – hodnotenie	69
D	Obsah priloženého pamäťového média	70

Kapitola 1

Úvod

Informačné systémy sa stali neoddeliteľnou súčasťou každodenného života ľudí a aj riadenia organizácií. Ako všetko, aj informačné systémy sa v priebehu času vyvíjajú, no zároveň aj zastarávajú. Tento prípad nastal aj v slovenskej firme EKOM spol. s r.o., ktorá potrebuje nahradiť dlho používaný systém novým riešením. Konkrétne sa jedná o informačný systém určený na spracovávanie dát procesmi workflow. Proces workflow, alebo podnikový proces, predstavuje sekvenciu krokov pre dosiahnutie určitého cieľu, pričom sa medzi týmito krokmi presúvajú dáta.

Väčšina výrobkov tejto firmy sú zdravotnícke zariadenia, ktoré musia podliehať prísnyim smerniciam a normám. Cieľom tejto práce je navrhnuť nový informačný systém, ktorý bude spracovávať dáta pre účely naplnenia týchto dokumentov a v súlade s požiadavkami firmy, no umožní aj vykonávanie procesov pre vnútorné účely firmy. Procesy majú byť definované štandardizovanými prostriedkami.

V tejto práci sa v kapitole 2 rozoberajú procesy workflow, ich spôsoby návrhu, riadenia a viaceré spôsoby, ktorými sa procesy dajú formálne zapísať. Pozornosť je postupne venovaná Petriho sieťam, sieťam procesov workflow a jazykom YAWL a BPMN.

Kapitola 3 sa zaoberá informačnými systémami ako takými, čo je ich úlohou, aké typy informačných systémov sa rozlišujú a aké sú možnosti ich architektonického rozdelenia. Ďalej sú tu popísané rôzne bežne používané možnosti komunikácií medzi aplikáciami a záver kapitoly je venovaný užitočným nástrojom.

V kapitole 4 je analyzované súčasné riešenie používané vo firme. Sú tu spomenuté požiadavky firmy na nový systém a taktiež dotazník pre používateľov aktuálneho systému, prostredníctvom ktorého sa zisťovali názory a požiadavky zamestnancov.

Po zoznámení sa so všetkými potrebnými informáciami nasleduje návrh nového riešenia v kapitole 5. Je tu popísaná predstava o fungovaní nového systému, spolupráca s ďalšími softvérmi a ukážka navrhnutých pohľadov v novom systéme.

Nasleduje implementácia navrhovaného systému, ktorá je popísaná v kapitole 6. Riešia sa tu jednotlivé súčasti celého systému a taktiež komunikácia medzi nimi. Základ nového informačného systému Workflow je implementovaný v podobe trojvrstvej architektúry, no okrem toho je dopĺňaný aj ďalšími softvérmi, s ktorými spolupracuje.

V poslednej kapitole 7 je popis prezentácie nového riešenia vo firme a testovanie s vybranými zamestnancami, pre ktorých má tento informačný systém kľúčový význam.

Kapitola 2

Procesy workflow, spôsoby ich definície a zápisu

Pojem *Workflow* označuje kolekciu usporiadaných úloh, ktoré vedú k vykonaniu určitého procesu. Tieto úlohy môžu byť vykonávané človekom, softvérom alebo ako kombinácia oboch [13]. Na prvý pohľad to môže pripomínať zoznam potrebných aktivít, ktoré sú však viac-menej nesúvisiace a poradie ich vykonania nie je veľmi dôležité. Workflow sa líši tým, že medzi jednotlivými krokmi presúva dáta, ktoré sa ďalej spracovávajú [24].

Podobnú definíciu má však aj *podnikový proces*. Podnikový proces pozostáva zo skupiny koordinovane vykonávaných činností, ktoré uskutočňujú podnikateľský cieľ. M. Weske potom označuje Workflow ako automatizáciu týchto procesov, kedy sa dokumenty, informácie alebo úlohy presúvajú medzi účastníkmi v súlade s danými procedurálnymi pravidlami [36]. Nakoľko sa jedná o dva súvisiace a podobné termíny, môžu byť vzájomne zameniteľné.

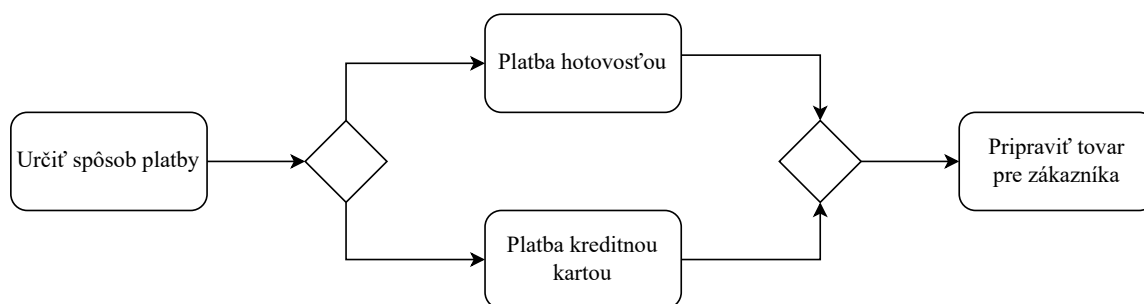
Vývoj problematiky workflow je podrobne popísaný od A. Sharp v [28], odkiaľ boli prevzaté najvýznamnejšie historické momenty. Počiatky vážneho vnímania podnikových procesov by sa dali datovať až do doby pred priemyselnou revolúciou, ktorá začala v polovici 18. storočia. Dovtedy väčšina produkcie prebiehala formou remesla, kde každý remeselník vykonával iba jednu konkrétnu činnosť, napríklad tkáč, kováč alebo klenotník. V rámci svojich aktivít riešili všetky fázy výroby až po dokončenie finálneho produktu. Okrem toho si sami riešili marketing, predaj, vývoj nových výrobkov, ako aj servis už vyrobených produktov. Toto obdobie bolo prínosné pre priamu komunikáciu medzi zákazníkom a výrobcom, kedy bolo veľmi jednoduché upraviť výrobky podľa osobných preferencií bez nedorozumení počas komunikácie. To so sebou však nieslo obmedzenú dostupnosť produktov, obtiažne škálovanie výroby alebo zdĺhavé zaúčanie nových ľudí. Priemyselná revolúcia, okrem iného, priniesla delbu práce na špecializované úlohy, čo zvýšilo efektivitu výroby nečakaným spôsobom. Príkladom môže byť rozdelenie výroby špendlíku na jednotlivé podúlohy, kde jedna osoba vykonávala iba jeden krok, čo dokázalo zvýšiť dennú produkciu až o 240 %. Väčším prínosom ako parný stroj bolo teda rozdelenie komplexnej práce na jednoduchšie úlohy a zúženie pracovných špecializácií. Zmeny sa nedotkli iba výroby ako takej, ale aj oblastí okolo nej, napríklad financie, účtovníctvo alebo právne záležitosti. Zaradením ľudí do skupín podľa zamerania práce opäť zlepšilo organizáciu, čo v 20. storočí umožnilo vznik veľkých korporácií. Pracovníci i oddelenia sa všeobecne snažia optimalizovať svoje činnosti v prospech organizácie. Ale optimalizácia časti nemusí vždy znamenať optimalizáciu celku. Je možné, že kvôli neznalosti celého procesu, optimalizácia časti poškodí celkový proces a jeho výsledok. Toto je jeden z neduhov špecializácie – zameranie sa iba na jednu časť bez znalosti celku.

Začiatkom 90. rokov 20. storočia prišiel Michael Hammer¹ s pojmom *business process reengineering*, teda reinžiniering podnikových procesov, ktorým priniesol nový spôsob myslenia do vtedajšej frustrovanej spoločnosti, vďaka čomu sa okamžite ujal. Pojem sa začal používať tak často a v nevhodných kontextoch, že sa stratil jeho pôvodný účel. Reinžinierovalo sa všetko. Aj to, čo nemalo s procesmi nič spoločné. Množstvom silného odporu voči reinžinieringu a rozšírením informačných technológií v druhej polovici 90. rokov, sa pozornosť presunula práve na informačné technológie a procesy ako také prišli o hlavnú pozornosť. Do popredia sa dostal internet a informačné systémy podnikov, čo malo za dôsledok nedodržiavanie podnikových procesov dostatočne vážne. Po roku 2000 nastalo zlúčenie neustáleho zlepšovania a reinžinieringu za využitia informačných technológií do spoločného pojmu, a to „riadenie procesov“. U dnešných podnikoch prevláda procesne orientovaný manažment so zameraním na výsledok pre zákazníka.

2.1 Moderný podnikový proces

Vstupom podnikových procesov sú zvyčajne dáta (získané napríklad z formuláru), ktoré sa v rámci ich aktivít spracovávajú presne špecifikovaným postupom. Spracované dáta z jednej aktivity sú vstupom pre nasledujúcu aktivitu [15]. Procesy workflow sú vhodné pre opakujúce sa procesy a úlohy v podnikoch, ktoré treba vykonávať v špecifickom poradí.

Príkladom takého procesu môže byť proces spracovania platby za objednávku. Príklad je prevzatý z [25]. Na obrázku 2.1 je zobrazený jednoduchý model procesu zapísaný v zjednodušenej podobe štandardu BPMN². Prvou aktivitou je *Určiť spôsob platby*, kedy systém obdrží dáta, ktoré obsahujú aj zvolený spôsob platby zákazníka. Následne sa proces v uzle v tvare kosoštvorca bez označenia rozhodne, ktorou vetvou bude pokračovať. Vo vrchnej vetve bude riešiť platbu hotovosťou v aktivite *Platba hotovosťou*, zatiaľčo v spodnej vetve rieši platbu kreditnou kartou v aktivite *Platba kreditnou kartou*. Vetvy sa spájajú v druhom uzle v tvare kosoštvorca bez označenia a proces ďalej pokračuje, keď sa dokončí buď jedna, alebo druhá možnosť platby. Na záver je pre zákazníka pripravený nakupovaný tovar v aktivite *Pripraviť tovar pre zákazníka* a po dokonaní končí aj samotný proces.



Obr. 2.1: Model jednoduchého procesu spracovania platby za objednávku. (Prevzaté a upravené z [25].)

¹Autor článku „Reengineering Work: Don’t Automate, Obliterate“, kde chcel napraviť priemyselnú revolúciu zmiernením špecializácie a opätovným zjednotením úloh do súvislých a zrejmych procesov

²BPMN – *Business Process Model and Notation*, viac na <https://www.bpmn.org/>

2.2 Typy podnikových procesov

Procesy v organizáciách možno kategorizovať podľa ich typu. Túto kategorizáciu [13] tvoria napríklad *materiálové* a *informačné* procesy. Úlohou materiálových je skladanie komponentov a vytváranie fyzických produktov (napríklad presun, skladovanie, transformovanie alebo skladanie fyzických objektov). V porovnaní s nimi sa informačné procesy zaoberajú automatizovanými a čiastočne automatizovanými úlohami (napríklad transakčné spracovávanie databázových operácií alebo distribuované systémy). Z pohľadu [24] sa podnikové procesy rozlišujú napríklad na *procesné*, *prípadové* alebo *projektové*. Procesné typy sa skladajú z predvídateľných a opakujúcich sa postupností krokov, zatiaľčo pri prípadových typoch nie je konkrétna postupnosť krokov na začiatku známa a závisí teda prípad od prípadu. Projektové procesy sa štrukturálne podobajú procesným, pridávajú však flexibilitu v tom, či vôbec a za akých okolností musia jednotlivé kroky nastať.

2.3 Návrh podnikového procesu

Procesy sa vyskytujú v podnikoch aj keď nie sú vhodne definované, či spravované. Také procesy sú však veľmi neefektívne. Premyslený návrh dokáže zlepšiť efektivitu operácií, zvýšiť spokojnosť zákazníkov, kvalitu služieb a znížiť náklady. Taktiež znižuje náchylnosť na chyby a iteratívnymi spätnými väzbami a úpravami sa môžu tieto procesy ďalej zlepšovať [24].

Koncept procesu workflow teda možno chápať aj ako návrh a automatizáciu podnikových procesov, ktoré sú konceptuálne popísané pre pochopenie, vyhodnotenie a úpravu.

Návrh zahŕňa nasledujúce kroky [24]:

1. Určenie počiatočného a koncového bodu procesu.
2. Určenie krokov potrebných na prechod z počiatočného bodu do koncového.
3. Zhodnotenie, či tieto kroky musia nasledovať v špecifickom poradí a ich zoradenie podľa potreby.
4. Určenie potrebných zdrojov a funkcií potrebných pre vykonanie individuálnych krokov a taktiež určenie podmienok alebo popis procesu.
5. Vykonanie procesu.

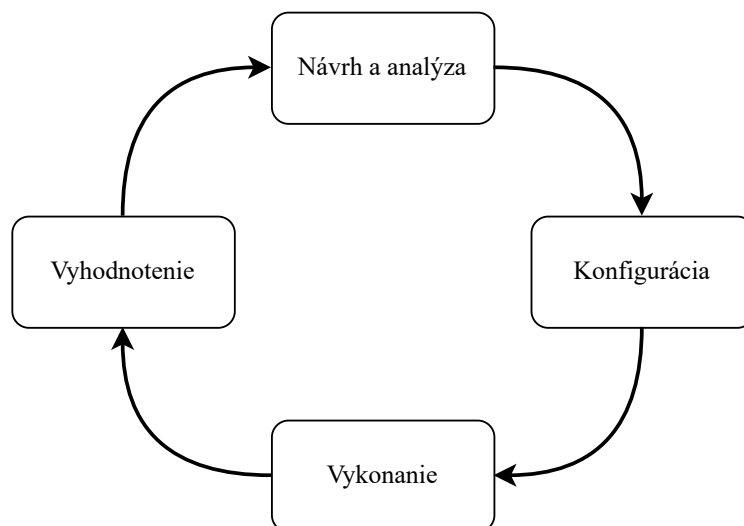
Alternatívy postup ponúka [28]. Návrh podnikového procesu je plán reakcie na určitú spúšťačiu udalosť. Určuje postupnosť krokov a rozhodnutí vykonaných medzi počiatočnou udalosťou a konečným výsledkom. V tomto prípade sa vykonávajú tri fázy. V prvej fáze sa určí rámec procesu. Identifikujú sa súvisiace procesy vytvorením celkovej mapy procesov a tým sa ohraničia ich rozsahy a styčné body. Zhodnotia sa zámery spoločnosti a určia sa ciele procesu v tomto kontexte. Druhá fáza sa zameriava na pochopenie aktuálneho procesu a prečo nedosahuje želané ciele. Dokumentuje kto, čo a kedy robí formou diagramov, pričom sa zameriava na celkový tok. Úroveň detailu popisu sa môže zvyšovať, kým správanie procesu nebude dostatočne jasné. Je vhodné označiť silné a slabé stránky aktuálneho procesu, ako aj vhodné body na výrazné zlepšenie. V poslednej fáze sa navrhuje nový proces s aplikovaním znalostí z predchádzajúcich fáz. Popisuje sa nový proces a čo sa mení v porovnaní s aktuálnym procesom. Navrhne sa konceptuálny model procesu a prechádzaním rôznymi úrovňami detailov a iteráciami sa odlaďujú detaily.

2.4 Riadenie procesov workflow

Podobne ako súvisí proces workflow s podnikovým procesom, súvisí aj ich riadenie. Riadenie podnikového procesu zahŕňa pojmy, metódy a techniky pre návrh, správu, konfiguráciu, zavedenie a analýzu podnikových procesov. Riadenie procesov workflow je reprezentácia štruktúr modelov procesov a kontrolované zavedenie podnikových procesov podľa týchto modelov [36].

Organizácia, ktorá si navrhne proces workflow, by sa o neho mala starať – od vytvorenia cez dokumentáciu a analýzu behu až po úpravy a vylepšenia. Vďaka tomu sa proces workflow môže optimalizovať pre konkrétne potreby a zabezpečiť sa správnosť, konzistencia a efektivita vykonania každého kroku. Vhodná správa taktiež odhalí problémy, prekážky a zbytočné kroky, ktoré môžu spomaľovať vykonávanie krokov a zvyšovať riziko chýb. Ich odstránením sa teda šetrí čas i peniaze. Následnou automatizáciou krokov alebo celých procesov workflow sa môžu dosiahnuť ďalšie benefity, napríklad rýchlejšie vykonávanie procesov workflow alebo minimalizovanie chýb manuálnym vykonávaním operácií [24].

Životný cyklus podnikového procesu sa dá vyjadriť schémou na obrázku 2.2, ktorá sa skladá zo štyroch fáz – *Návrh a analýza*, *Konfigurácia*, *Vykonanie* a *Vyhodnotenie*. Prvou fázou je *Návrh a analýza*, kde sa identifikujú a zhodnocujú informácie o procese, ktorý je následne reprezentovaný modelom podnikového procesu. Model zapísaný v grafickej podobe uľahčuje zainteresovaným osobám z rôznych oddelení validáciu všetkých možných scenárov a doladenie chýb. Výstupom tejto fázy je formálny popis v konkrétnej podobe zápisu modelovania podnikového procesu. V nasledujúcej fáze, *Konfigurácia*, sa pripravuje prostredie pre beh procesu – interakcia s používateľmi a inými systémami, ale aj implementácia procesu s overením funkčnosti. Integrované a výkonnostné testy sú tu tiež vykonávané, aby sa odhalili problémy, ktoré by mohli nastať za behu procesu. Záverom tejto fázy je proces nasadený do prevádzky. Fáza *Vykonanie* zahŕňa samotné vykonávanie a sledovanie stavu navrhnutého procesu systémom pre správu podnikových procesov podľa definovaného modelu. Systém okrem toho zbiera rôzne dáta, zvyčajne vo forme záznamov, ktoré opisujú priebeh vykonávania procesov, ako napríklad začiatok a ukončenie jednej aktivity. Tieto záznamy sú základom pre ďalšiu fázu. Vo fáze *Vyhodnotenie* sa analyzujú záznamy z behu procesov a získané informácie sa potom použijú v ďalšej iterácii úpravy procesu [36]. Ako je vidieť na obrázku 2.2, životný cyklus procesu beží v kruhu bez koncového bodu. To značí neustálu potrebu zlepšovať a inovovať procesy v dynamicky sa meniacej podnikovej sfére.



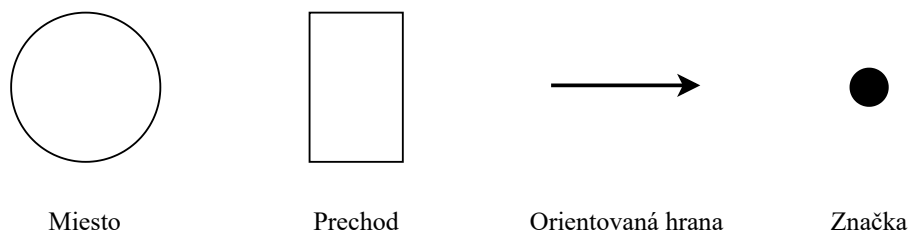
Obr. 2.2: Schéma životného cyklu podnikového procesu. (Inšpirované [36].)

2.5 Spôsobý zápisu modelov podnikových procesov

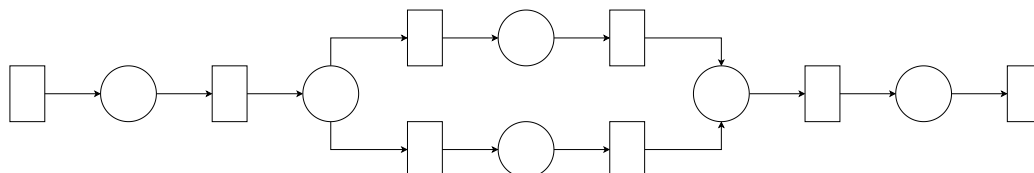
Modely podnikových procesov určujú aktivity a ich vzájomné vzťahy, ktoré sú vykonávané v rámci organizácie a systém na riadenie podnikových procesov sa riadi týmito modelmi. Pre reprezentáciu modelov existuje viacero riešení. Jednou z možností sú všeobecné *Petriho siete*. Alternatívou sú *siete procesov workflow* špeciálne určené pre modely podnikových procesov. Tie sú vhodné pre modelovanie a analýzu procesov workflow, ale majú svoje obmedzenia, ktoré viedli k vývinu modelovacieho jazyka *YAWL* (Yet Another Workflow Language). Ďalším štandardom, ktorý využíva jasný grafický zápis a sémantiku, je *BPMN* (Business Process Modeling Notation) [36].

Petriho siete

Petriho siete predstavujú prístup k zápisu procesov, ktorý je založený na formalizme modelovania a analýzy procesov. Použitie formálnych prostriedkov má oproti neformálnym technikám založeným na diagramoch niekoľko výhod – vynucuje presné definície, potláča nejasnosti a sporné situácie. Petriho siete však stále umožňujú vizuálnu reprezentáciu procesov, ktorá má formalizovaný matematický základ. Petriho siete pozostávajú z prvkov označujúcich miesta, prechody a orientované hrany spájajúce miesta a hrany. Miesta predstavujú stav v rámci systému a prechody predstavujú aktivity alebo udalosti, ktorými sa stav mení. Ich zápis nie je štandardizovaný, no bežne používaný zápis je zobrazený na obrázku 2.3, ktorý zľava doprava ukazuje tieto prvky: miesto, prechod, orientovaná hrana, značka. Prechody majú vstupné a výstupné miesta. Zo vstupných miest vedie hrana končiacia na prechode a z prechodu vedie hrana smerujúca na výstupné miesta. Dynamika systému je určená pozíciou značiek, ktoré sa podľa určitých pravidiel presúvajú po prechodoch medzi miestami. Aktuálna pozícia značiek určuje aktuálny stav Petriho siete a teda aj modelovaného systému. Obrázok 2.4 ukazuje proces spracovania platby za objednávku predstavený v časti 2.1. Petriho siete teda reprezentujú model procesu a značky inštanciu tohto procesu. Nakoľko sa značky v klasických Petriho sieťach nedajú odlišiť, môže na modeli naraz existovať iba jedna inštancia [2].



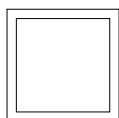
Obr. 2.3: Prvky používané v Petriho sieťach.



Obr. 2.4: Proces spracovania platby za objednávku zapísaný v podobe Petriho sietí.

Siete procesov workflow

Ako bolo naznačené v úvode tejto sekcie, siete procesov workflow sú rozšírením Petriho sietí. Petriho siete sú vhodné pre jednoduché procesy, ale neposkytujú mechanizmy pre zložitejšie procesy, akými sú procesy workflow. Podobne ako u Petriho sietí, miesta označujú podmienky, značky reprezentujú inštancie a prechody predstavujú aktivity. V porovnaní s Petriho sieťami, značky v sieťach procesov workflow obsahujú dáta o konkrétnej inštancii procesu vrátane jej identifikátoru, napríklad zafarbenie značky. Okrem toho umožňujú vnorenie procesov do seba alebo explicitné označovanie prechodov pri vetvení. Spôsob zápisu vnoreného procesu je vidieť na obrázku 2.5. Tento vnorený proces je ďalším modelom siete procesov workflow [2].



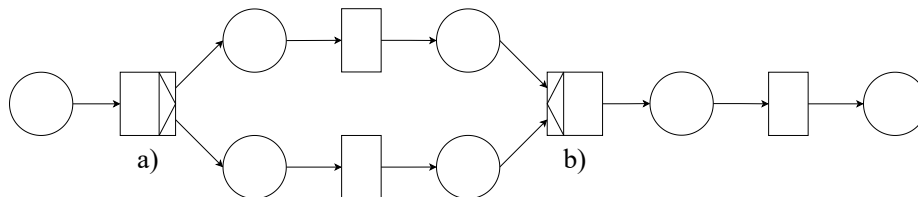
Obr. 2.5: Zložená aktivita reprezentujúca ďalší model siete procesov workflow. (Prevzaté z [2].)

Siete procesov workflow taktiež prinášajú nasledujúce štrukturálne obmedzenia [2]:

- Model má iba jedno začiatkové miesto (bez vstupných hrán).
- Model má iba jedno koncové miesto (bez výstupných hrán).
- Každé miesto a každý prechod leží na ceste zo začiatkového miesta do koncového.
- Po dokončení inštancie procesu nesmie v procese zostať žiadna značka.
- Pre každú jednu značku v začiatkovom mieste skončí v koncovom mieste jedna značka.

Obrázok 2.6 zobrazuje proces spracovania platby za objednávku rovnako ako pri klasických Petriho sieťach. V tomto prípade sú rozdielne notácie vetviacich prvkov. Prvok

označený písmenkom a) označuje rozdelenie vetví logickou spojkou *XOR* a prvok b) zasa komplementárne spojenie vetví rovnakou spojkou.

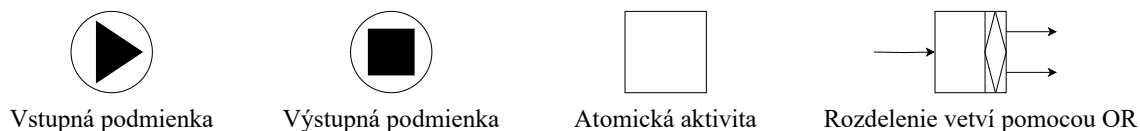


Obr. 2.6: Proces spracovania platby za objednávku v podobe sietí procesov workflow.

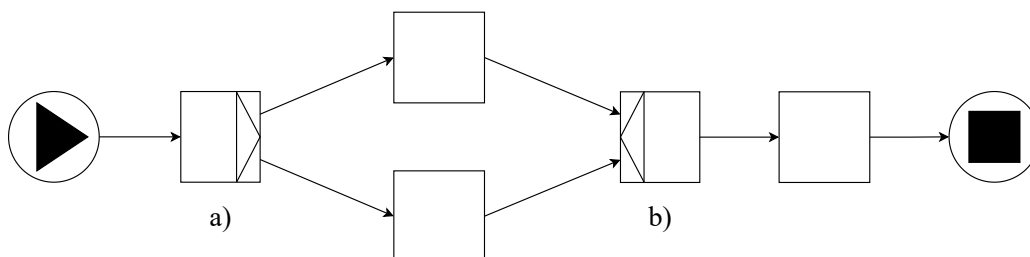
Yet Another Workflow Language

Yet Another Workflow Language (YAWL), voľne preložiteľný ako „Opäť ďalší jazyk pre procesy workflow“, vznikol kvôli neexistencii modelovacieho jazyka procesov, ktorý by pokryl všetky vzory toku riadenia. Jazyk YAWL syntakticky vychádza zo sietí procesov workflow, no líši sa v sémantike vykonávania inštancií procesov. Rozširuje siete procesov workflow o možnosť priamych spojení orientovanými hranami medzi prechodmi či anotovanie vetviacich a spájajúcich prechodov konkrétnym správaním. Popritom upravuje alebo dopĺňa symboly sietí procesov workflow. Začiatkové a koncové miesta majú nový vzhľad, pribudla možnosť viacerých inštancií aktivít a zaviedlo sa vetvenie pomocou logickej spojky *OR* [3].

Procesy workflow potrebujú dodatočné informácie pre vyhodnocovanie podmienok. Napríklad pre rozhodnutie cesty pri vetvení logickou spojkou *OR* alebo *XOR* treba vedieť podmienku, podľa ktorej sa vyberie konkrétna vetva [1]. Vybrané symboly jazyka YAWL možno vidieť na obrázku 2.7 a príklad použitia na už známom procese spracovania platby za objednávku na obrázku 2.8. V jazyku YAWL sú prvky pre rozdeľovanie (a)), respektíve spájanie (b)), vetiev spojkou *XOR* rovnaké ako pri sieťach procesov workflow.



Obr. 2.7: Symboly jazyka YAWL. (Pôvodom z [3].)



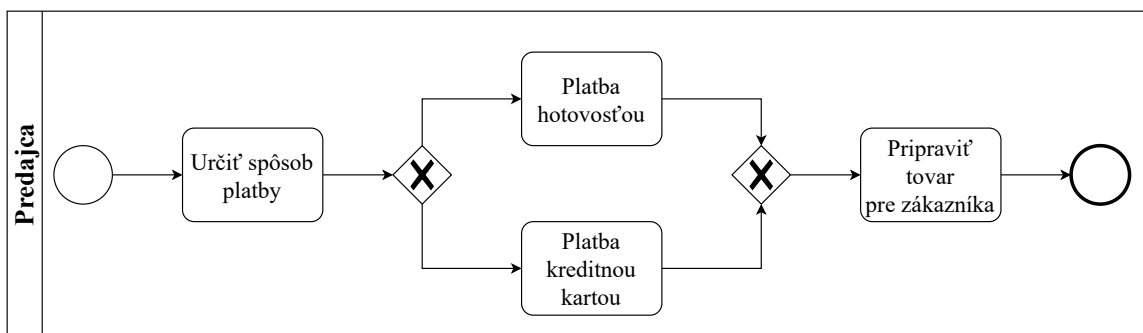
Obr. 2.8: Proces spracovania platby za objednávku zapísaný v jazyku YAWL.

Business Process Modeling Notation

Jazyk Business Process Modeling Notation (BPMN) bol vyvinutý organizáciou Object Management Group (OMG)³ a schválený ako štandard ISO popri ďalších štandardoch tejto organizácie ako napríklad Unified Modeling Language (UML)⁴ alebo Object Constraint Language (OCL)⁵.

Úloha jazyka BPMN pri modelovaní podnikových procesov je veľmi podobná úlohe jazyka UML pri objektovo-orientovanom návrhu, teda prevziať osvedčené postupy existujúcich prístupov a skombinovať ich do nového a všeobecne akceptovaného jazyka. Tento jazyk sa inšpiruje prvkami zo sietí procesov workflow založených na Petriho sieťach a tiež jazykom UML, konkrétne diagramom aktivít [36]. V špecifikácii [21] sa píše: „Primárnym cieľom jazyka BPMN je poskytnúť zápis, ktorý je ľahko zrozumiteľný všetkým používateľom z biznisu, od podnikových analytikov, ktorí vytvárajú počiatkové návrhy procesov, až po technických vývojárov zodpovedných za implementáciu technológií, ktoré budú tieto procesy vykonávať, a nakoniec obchodníkom, ktorí budú tieto procesy riadiť a monitorovať. Jazyk BPMN tak vytvára štandardizované prepojenie medzi návrhom podnikových procesov a implementáciou procesov.“

Modely podnikových procesov majú podobu diagramov podnikových procesov a skladajú sa z pevne definovanej množiny modelovacích prvkov [25]. Modelovacie prvky možno rozdeliť na základnú množinu a úplnú množinu prvkov. Prvky zo základnej množiny sú dostatočné na vyjadrenie jednoduchých konštrukcií a zároveň ľahké na pochopenie bez rozsiahlej znalosti tohto jazyka. Vyjadrovaciu schopnosť možno následne zlepšiť využitím prvkov z úplnej množiny, čo však vyžaduje hlbšie zoznámenie sa s týmto jazykom [21]. Príklad zápisu jednoduchého procesu spracovania platby za objednávku je vidieť na obrázku 2.9. V porovnaní so zápisom na obrázku 2.1 predstavenom v úvode kapitoly, je tento proces doplnený o symboly začiatkovej udalosti – jednoduchý kruh a koncovkej udalosti – kruh s hrubým okrajom. Taktiež obsahuje explicitné označenie brán a explicitné pomenovanie vykonávateľa tohto procesu, „Predajca“, ktorý prebieha v jednej plaveckej dráhe. Príklad ukazuje základné typy prvkov jazyka BPMN: udalosti, aktivity, brány a sekvenčný tok.



Obr. 2.9: Proces spracovania platby za objednávku zapísaný v jazyku BPMN.

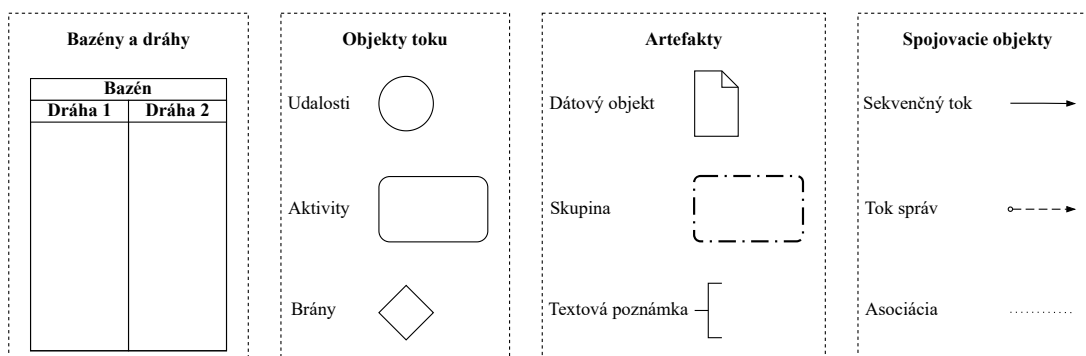
Organizačne možno rozdeliť procesy do *bazénov* a *dráh*. Grafická reprezentácia (súčasť obrázku 2.10) pripomína plavecký bazén s plaveckými dráhami, a preto sa používajú aj tieto pomenovania. Každý proces sa nachádza iba v jednom bazéne, nakoľko sa vykonáva iba v rámci jedného podniku. Pre interakciu s iným podnikom sa modeluje iný bazén, ktorý

³Domovská stránka organizácie OMG: <https://www.omg.org/>

⁴Špecifikácia jazyka UML: <https://www.omg.org/spec/UML>

⁵Špecifikácia jazyka OCL: <https://www.omg.org/spec/OCL>

reprezentuje tento podnik a samotná interakcia prebieha zasielaním správ. Ak sa v jednom procese vyskytuje viacero rolí vykonávateľov, tieto role sú reprezentované dráhami v jednom bazéne. Naprieč dráhami sa môže voľne prechádzať sekvenčným tokom, nakoľko sa jedná o jeden proces v jednej organizácii. Na obrázku 2.10 možno vidieť aj *objekty toku*. Relevantné stavy reálneho sveta sú reprezentované *udalosťami*, vykonávaná práca počas procesov je zastúpená *aktivitami*, ktoré môžu byť atomické alebo zložené, a *brány* slúžia na rozdeľovanie a spájanie sekvenčných tokov. Ďalšou kategóriou na obrázku 2.10 sú *artefakty*. Tie poskytujú dodatočné informácie o podnikovom procese, ktoré však majú iba dokumentačný charakter a neovplyvňujú priebeh samotného procesu. *Dátový objekt* označuje použitie alebo vytvorenie dátového objektu pod nejakým názvom na hocijakom médiu. Textová *poznámka* pridáva popis iného prvku, ku ktorému je priamo pripojená. *Skupina* označuje prvky procesu patriace do rovnakej skupiny, čo nemá iný význam než sprehľadnenie modelu. Poslednou kategóriou na obrázku 2.10 sú *spojovacie objekty*. Tie prepájajú prvky z kategórií popisovaných vyššie. *Sekvenčný tok* určuje zoradenie vykonávania *objektov toku*. *Tok správ* zobrazuje zasielanie správ medzi obchodnými partnermi (reprezentovanými bazénmi) a *asociácia* pridružuje *artefakty* k iným prvkom [36].

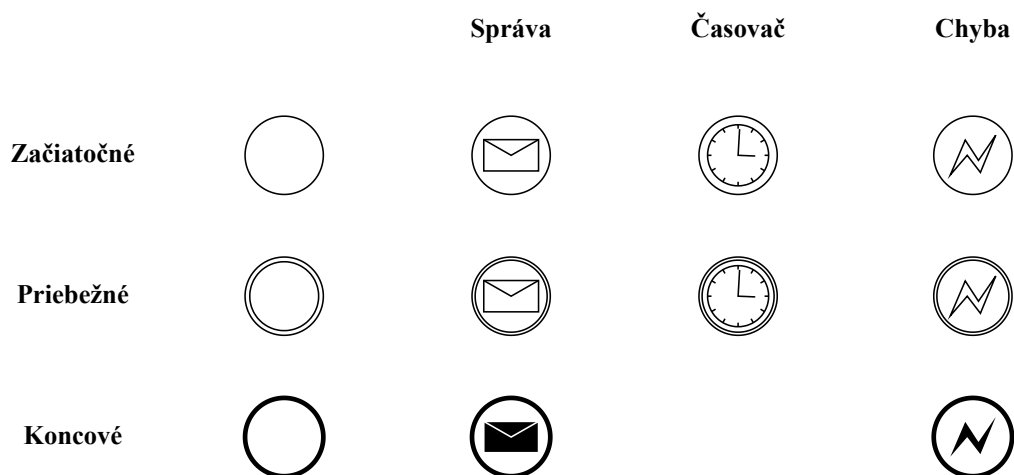


Obr. 2.10: Kategórie prvkov jazyka BPMN. (Prevzaté z [36].)

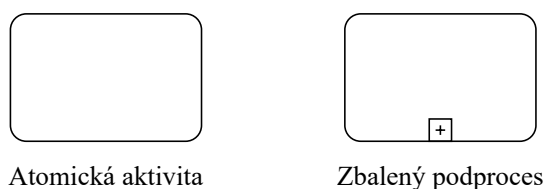
Udalosti sú prechodovými bodmi medzi stavmi v podnikoch a procesmi reagujúcimi na výskyt týchto udalostí. Udalosti sa dajú rozdeliť na tri typy – *začiatkové* (spúšťajú proces), *priebežné* (nastávajú počas procesu) a *koncové* (ukončujú proces). Každý typ môže mať viacero variantov, ktoré sú špecifické doplnujúcim symbolom vnútri. Príklady vybraných variantov udalostí sú na obrázku 2.11. Ich popis, ako aj ďalšie varianty, sú detailne popísané v [21].

Aktivity sú samotné položky práce v podnikovom procese, ktoré sú hierarchicky vnorené, takže každá aktivita je buď atomická aktivita (obrázok 2.12 vľavo) alebo podproces (obrázok 2.12 vpravo). Keď vnútorná štruktúra aktivity už nie je dôležitá, ide o atomickú aktivitu. Na najvyššej úrovni je proces sám osebe [21]. Rozpisovaním na podprocesy sa dá dosiahnuť zápis na požadovanej úrovni abstrakcie. Z pohľadu riadenia procesov môže stačiť abstraktnejší pohľad, no pre účel implementácie je užitočné proces detailnejšie rozpísať. Osvedčeným postupom je ponechať každú aktivitu iba s jednou vstupnou hranou a jednou výstupnou, aby sa zachovala jednoúčelnosť každého elementu [36].

Brána môže slúžiť buď ako rozdeľovací alebo spájajúci uzol. Pre oba typy sa používa rovnaký symbol a odlišiť sa dajú nasledovne. Rozdeľovací uzol má práve jednu vstupnú hranu a aspoň dve výstupné hrany. Spájajúci uzol to má presne naopak – aspoň dve vstupné hrany a práve jednu výstupnú hranu. Ukážky symbolov brán používajúcich logické spojky XOR (výlučná), OR (zahrňujúca), AND (paralelná) sú na obrázku 2.13. Pre výstupné hrany

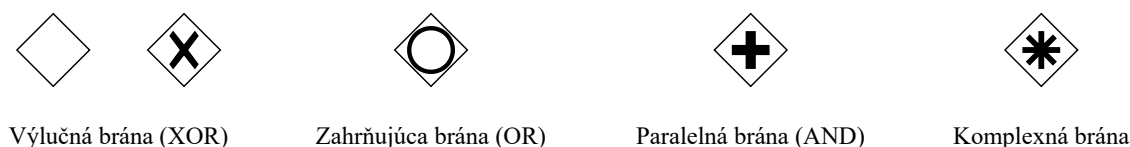


Obr. 2.11: Príklad rôznych udalostí v jazyku BPMN. (Prevzaté z [21].)



Obr. 2.12: Prvky označujúce atomickú aktivitu a podproces. (Prevzaté z [21].)

rozdeľovacích brán je vhodné definovať podmienky, za ktorých sa zvolí tá či oná vetva. Volí sa tá vetva, ktorej pridružená podmienka je vyhodnotená ako pravdivá. Je možné definovať aj predvolenú (implicitnú) vetvu pre prípad, keby všetky ostatné podmienky boli vyhodnotené ako nepravdivé. Rozdeľujúca *zahrňujúca brána* sa používa pre výber ľubovoľného počtu vetiev. Návrhár procesu musí zaistiť, že sa vykoná aspoň jedna vetva. *Paralelná brána* rozdeľuje tok do viacerých paralelných vetiev, ktoré sa musia všetky vykonať, aby mohol proces ďalej pokračovať. Poslednou bránou na obrázku 2.13 je *komplexná brána*. Tá slúži pre definovanie zložitejších výrazov, spojených s výstupnými hranami, na určenie výstupných vetiev [21]. Úloha spájajúcich uzlov je komplementárna k tým rozdeľovacím a určujú bod a typ synchronizácie viacerých vetiev.



Obr. 2.13: Ukážky brán v jazyku BPMN. (Prevzaté z [21].)

Elementy sekvenčného toku určujú poradie objektov toku v procese, pričom každý je reprezentovaný plnou šípkou medzi objektmi toku. Každý element sekvenčného toku má práve jeden začiatok a práve jeden cieľ, ktoré musia byť udalosť, aktivita alebo brána. Normálny tok označuje očakávané a želané správanie procesu, ktorý začína začiatkovou udalosťou, prechádza objektmi toku a končí v koncovkej udalosti [21].

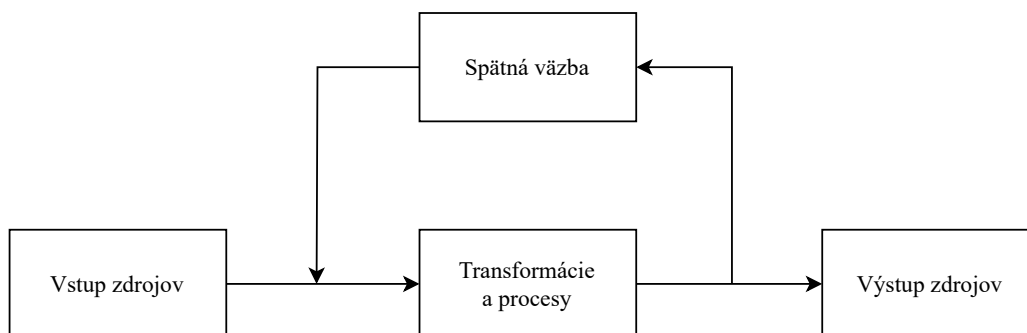
Kapitola 3

Informačné systémy a nástroje pre riadenie procesov

V predchádzajúcej kapitole boli predstavené procesy vykonávané v podnikoch, konkrétne procesy workflow, a možnosti, ako ich zapísať vo viacerých grafických podobách. Zjednotenie podnikových procesov ich grafickým zápisom je síce určitý krok k zlepšeniu, pracovníkom v podnikoch to však prácu veľmi neuľahčí. Tomu by mohlo pomôcť zavedenie systému, ktorý by vykonával tieto procesy a s pracovníkmi by interagoval v prípade potreby. Takýmto riešením býva zvyčajne informačný systém, na ktorý sa zameriava táto kapitola.

Pojem *informačný systém* je združené pomenovanie¹, teda slovné spojenie, ktoré ako celok vyjadruje jeden pojem, ale jeho komponenty majú svoj základný význam. Slovo *informačný* je odvodené od slova *informácia*, čo predstavuje interpretované dáta koncovým používateľom. Teda dáta, ktoré majú v určitom kontexte určitý význam. Zaradením týchto informácií do súvislostí vznikajú znalosti, inak povedané odvodené (sekundárne) informácie [14].

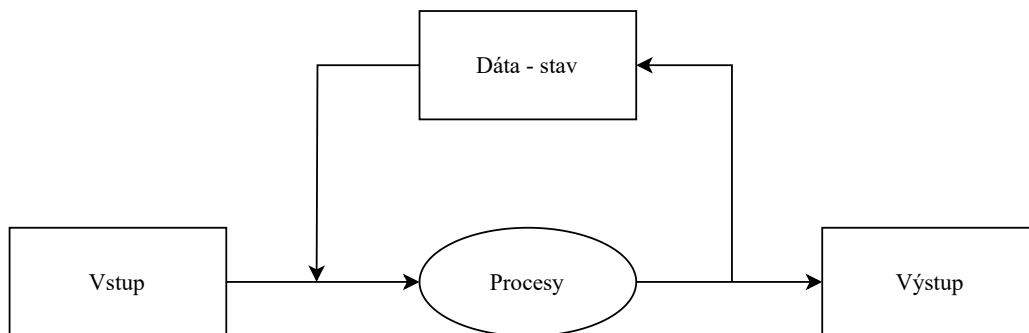
Druhou časťou pojmu *informačný systém* je *systém*. Systém (všeobecne) je množina prvkov a vzťahov medzi nimi, ktoré sú definované na nejakom nosiči. Nosičom je teda množina prvkov systému a ich vzájomné vzťahy. Jeho prvky sa nazývajú zdroje. Obrázok 3.1 zobrazuje schému všeobecného systému. Do systému zdroje vstupujú a taktiež z neho vystupujú. Medzi tým prebieha ich transformácia a popritom bežia systémové procesy. Systém väčšinou obsahuje aj spätnú väzbu, takže výstup systému závisí nielen na vstupných zdrojoch, ale aj na stave systému, ktorý predstavujú hodnoty zdrojov [14].



Obr. 3.1: Schéma všeobecného systému. (Prevzaté z [14].)

¹Podrobnejší popis združeného pomenovania možno nájsť na: juls.savba.sk

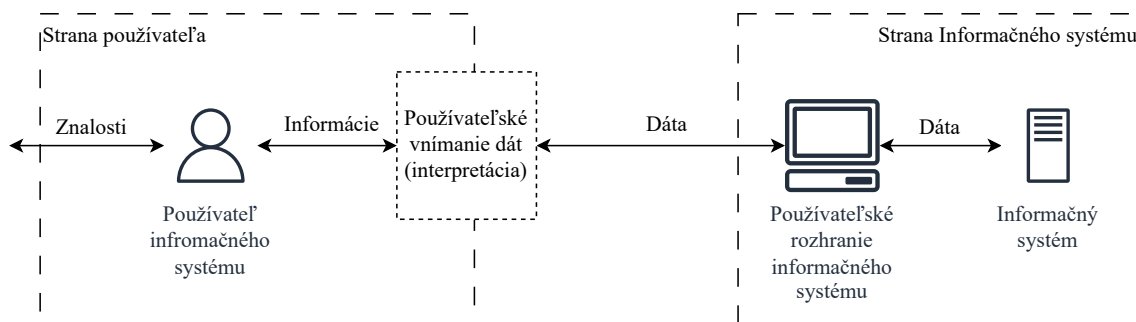
Spätným spojením do pojmu *informačný systém* dostávame systém ktorý spracováva dáta a jeho stav je tiež určený dátami. Spracovanie (transformácia) týchto dát prebieha medzi vstupom a výstupom formou transakcií, ktoré sú vykonávané v rámci procesov. Spätná väzba v tomto prípade predstavuje ukladanie stavu systému. Podobný pohľad na informačný systém má aj K. Laudon v [17], kde definuje informačný systém ako súbor vzájomne prepojených zložiek, ktoré zhromažďujú, spracovávajú, ukladajú a distribuujú informácie pre podporu rozhodovania v organizácií. Schéma všeobecného systému upravená na informačný systém je zobrazená na obrázku 3.2.



Obr. 3.2: Schéma informačného systému. (Prevzaté z [14].)

Nosič informačného systému reprezentuje nejaký skutočný systém. Tým pádom možno povedať, že samotný informačný systém je abstraktným modelom nejakého skutočného systému. Úlohou informačného systému nie je transformovať dáta na informácie. Informačný systém iba poskytuje dáta a na používateľoch je vytvoriť ich interpretáciu v danom kontexte [14]. Naopak, účelom informačného systému je zabezpečiť dostupnosť správnych dát v správny čas a na správnom mieste [8]. Obrázok 3.3 ukazuje komunikáciu medzi používateľom a informačným systémom. Informačný systém a k nemu pridružené používateľské rozhranie pracujú iba s dátami. Až keď používateľ tieto dáta interpretuje, stávajú sa pre neho informáciami, z ktorých následne môže odvodzovať ďalšie znalosti.

Nakoľko táto práca pojednáva o informačnom systéme pre konkrétnu firmu, pozornosť bude ďalej venovaná podnikovým informačným systémom.



Obr. 3.3: Presun dát od informačného systému k používateľovi.

Základom pre správne fungovanie podniku a dosahovanie strategických cieľov je efektívne spracovávanie informácií a budovanie báze znalostí, na čo posluží podnikový informačný systém. Petr Sodomka [29] definuje podnikový informačný systém nasledovne: „Podnikový informačný systém vytvárajú ľudia, ktorí prostredníctvom dostupných technologic-

kých prostriedkov a stanovenej metodiky spracovávajú podnikové dáta a vytvárajú z nich informačnú a znalostnú bazu organizácie slúžiacu k riadeniu podnikových procesov, manažérskeho rozhodovaniu a správe podnikovej agendy.“

Podobne ako je usporiadaná organizačná štruktúra podniku, možno hierarchicky usporiadať zameranie podnikových informačných systémov. Najčastejšie sa rozlišujú úrovne *strategická, riadiaca, znalostná a prevádzková*, ktoré sa dajú znázorniť pyramídovou schémou na obrázku 3.4. Zámerom tejto klasifikácie je určiť hodnotu automatizovaného spracovania informácií pre pracovníkov na danej organizačnej úrovni [29].



Obr. 3.4: Pyramídová schéma hierarchického usporiadania informačných systémov. (Inšpirované [29].)

Prevádzková úroveň (Transaction Processing Systems, skrátene TPS) sa zameriava na základné aktivity a transakcie podniku (napríklad predaj, výroba, platby, tok materiálu a podobne), ktoré sa vykonávajú na dennej báze. Hlavnou úlohou informačného systému tejto úrovne je poskytovanie odpovedí na rutinné otázky a sledovanie toku transakcií v organizácii, takže sa vyžaduje dostupnosť, aktuálnosť a presnosť potrebných informácií. Operácie prevádzkovej úrovne sú presne definované a štrukturované, a sú dôležitými zdrojmi informácií pre systémy na vyšších vrstvách. Systémy TPS sú pre podniky často kriticky dôležité a ich výpadok čo i len na niekoľko hodín môže spôsobiť vážne komplikácie [17].

Stredný manažment potrebuje monitorovať, kontrolovať a rozhodovať o administratívnych aktivitách. K tomu slúžia informačné systémy *znalostnej úrovne* (Management Information Systems, skrátene MIS), ktoré sa zameriavajú na to, či všetko funguje tak, ako má. Systémy MIS poskytujú hlásenia o aktuálnej výkonnosti organizácie, ktoré slúžia na predpovedanie tej budúcej. Hlásenia vznikajú na základe dát poskytnutých systémami TPS (nižšej úrovne) na pravidelnej báze (týždenne, mesačne alebo ročne). Tieto systémy odpovedajú na vopred špecifikované otázky za predurčeného postupu, vďaka čomu nie sú zrovna flexibilné a majú malé analytické možnosti, ktoré sú založené na zhrnutiach a porovnaníach [17].

Naopak, *riadiaca úroveň* (Decision-Support Systems, skrátene DSS) sleduje záležitosti, ktoré sú unikátne a rýchlo sa meniace, vďaka čomu sa obslužné procedúry nedajú vopred úplne definovať. Táto úroveň rieši otázky štýlu „Čo by sa stalo ak ...“ alebo „Ktoré riešenie je optimálne?“, pričom sa opiera o dáta z oboch nižších úrovní, ktoré dopĺňa o informácie z externých zdrojov, napríklad ceny podobných výrobkov konkurencie [17].

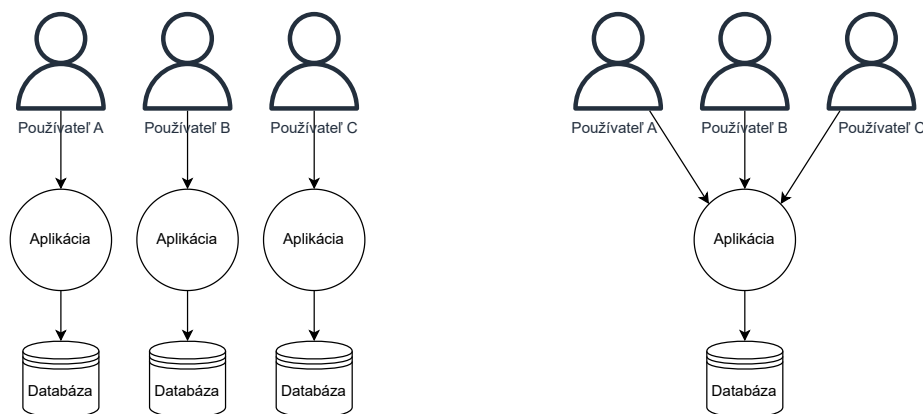
Strategická úroveň (Executive Support Systems, skrátene ESS) pomáha vrcholovému manažmentu s nevhodnými rozhodovaniami, ktoré vyžadujú posúdenie, zhodnotenie a prehľad, pretože neexistuje dohodnutý postup, ako riešiť tieto situácie. Systémy ESS zohľadňujú dlhodobé dáta o externých udalostiach, ktoré kombinujú so sumárnymi informáciami z interných systémov MIS a DSS. Pozornosť sa kladie na kriticky dôležité dáta, ktoré sú prezentované formou grafov na takzvaných „dashboard“ obrazovkách. Systémy ESS tiež analyzujú trendy a na základe nich ponúkajú budúce predpovede [17].

3.1 Architektúry informačných systémov

Výber vhodnej architektúry predstavuje jedno z kľúčových rozhodnutí pri vývoji informačného systému. Kvalitná architektúra dokáže predchádzať rade problémov, ktoré môžu nastať v priebehu prevádzky informačného systému. V prípade architektúry softvéru sa kladie dôraz na štruktúru a vzťahy komponentov, teda modulov, z ktorých sa skladá aplikácia. Architektúra určuje, či bude aplikácia pre jedného alebo viacerých používateľov; z kolkých modulov sa bude skladať a ako budú vzájomne usporiadané; o akú funkcionálnosť sa bude modul starať; alebo aplikačné programové rozhrania (*Application Programming Interface*, skrátene API) pre iné moduly či aplikácie. Kategorizácia architektúr pochádza z [8].

Architektúra s jedným používateľom predstavuje plnohodnotnú aplikáciu bežiacu individuálne na zariadení každého používateľa. Aplikácie tejto architektúry majú zvyčajne nižšie náklady na vytvorenie, no vyššie náklady prídu počas prevádzky vďaka vyšším nárokom na výkon zariadenia. V dnešnej dobe ide o takmer nevyužívanú architektúru, nakoľko vo väčšine prípadov je potrebná súčasná spolupráca viacerých používateľov s rovnakými dátami.

Aplikácia *architektúry s viacerými používateľmi* je schopná naraz obsluhovať viac, teoreticky až nekonečno, používateľov. Typickým znakom je, že všetci používatelia takejto aplikácie zdieľajú rovnaké dáta z rovnakého zdroja, teda databázy. Obrázok 3.5 ilustruje rozdiel medzi týmito dvoma architektúrami.

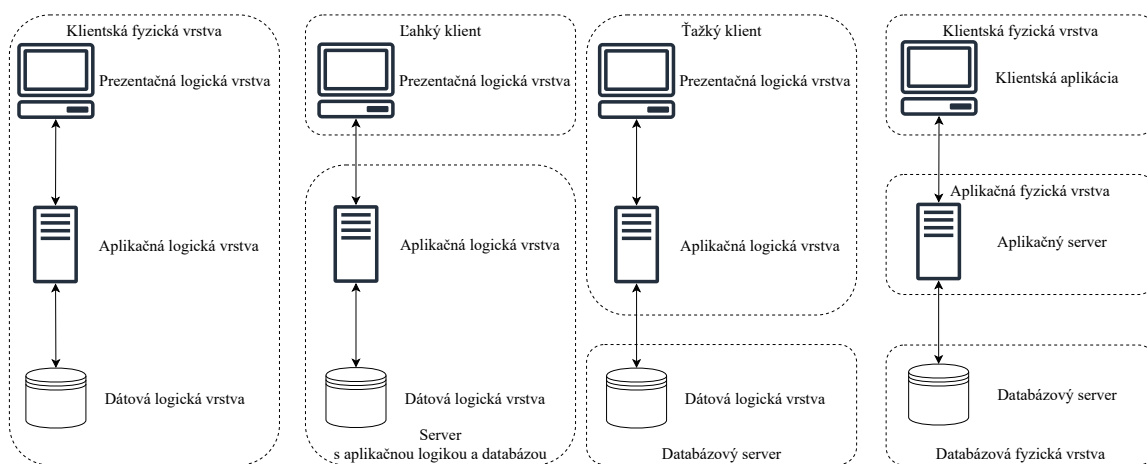


Obr. 3.5: Ukážka rozdelenia na jednopoužívateľské (vľavo) a viacpoužívateľské (vpravo) architektúry. (Inšpirované [23].)

Architektúra klient-server rozdeľuje aplikáciu na dva, prípadne viacero, spolupracujúce programy. Klient predstavuje tú časť aplikácie, ktorá žiada o vykonanie nejakej funkcionality. Naopak, server je program, ktorý túto funkcionálnosť poskytuje po vyžiadaní. V prípade viacerých spolupracujúcich programov môže každý z nich vystupovať v úlohe klienta aj servera podľa toho, ako medzi sebou komunikujú a kto od koho čo žiada. Výhodou tohto

prístupu sú nižšie nároky na server – čas výpočtovej záťaže sa prenáša na klienta riešiacoho používateľské rozhranie; tým pádom aj jednoduchšie škálovanie výpočtových kapacít či možnosť využitia rôznych platforiem.

Nevýhodou však môže byť vyššia počiatočná investícia (čas vývoju aj hardvérové nároky) alebo náročnejšie zabezpečenie korektnej komunikácie medzi časťami aplikácie. Aplikácie tejto architektúry sa bežne rozdeľujú na niekoľko fyzických vrstiev podľa toho, ako treba oddeliť základné skupiny funkcionalít. Funkcie bývajú bežne rozdelené na dátové funkcie (nakladanie s dátami na úrovni databázy; dátová vrstva), transformačné funkcie (samotná požadovaná biznis logika spracovávania dát; aplikačná vrstva) a komunikačné funkcie (zobrazovanie dát používateľom a komunikácia s nimi; prezentačná vrstva). Bežné možnosti rozdelenia sú na obrázku 3.6.



Obr. 3.6: Vizualizácia architektúr (zľava): monolitická, dvojvrstvová (lahký klient a ťažký klient) a trojvrstvová. (Inšpirované [26].)

Prvou variantou rozdelenia je *monolitická architektúra*, ktorá sa nedelí na časti programu, a tak sú všetky funkcie vykonávané v rámci jednej vrstvy jedným programom. Toto je typické pre centralizované spracovávanie, kde sú spomenuté skupiny funkcií navzájom prepletené. Výhodou je jednoduchšia ochrana funkcií a dát voči neautorizovanému prístupu. Problémy však nastávajú pri údržbe aplikácií, napríklad zmena databázy môže vyžadovať úpravu kódu na mnohých miestach, čo zvyšuje časovú náročnosť ako aj riziko chýb.

Druhou možnosťou je oddelenie jednej skupiny od zvyšných dvoch, čím vznikne *dvojvrstvová architektúra*. Zámerom je oddeliť komunikačné a dátové funkcie a podľa umiestnenia transformačných funkcií sa rozlišuje ľahký klient (prezentačná vrstva osamostatnená a aplikačná s dátovou spolu) alebo ťažký klient (prezentačná a aplikačná vrstva spolu, dátová vrstva osamostatnená). Týmto oddelením možno ľahšie realizovať viaceré formy komunikácie s rôznymi koncovými stanicami tým, že pre každý účel sa vytvorí samostatný program prezentačnej vrstvy (napríklad prispôbenie pre stolné počítače a mobilné zariadenia).

Ak sú všetky tri spomenuté skupiny rozdelené do samostatných programov, ide o *trojvrstvovú architektúru*. Vrstvy v rámci architektúry navzájom komunikujú princípom klient-server popísaným vyššie. Všetky vrstvy sa dajú udržiavať a vyvíjať samostatne, a teda ich plne optimalizovať pre konkrétny účel. Toto je vhodným riešením pre flexibilné a ľahko prispôbitelné informačné systémy. Realizovať zmenu kvôli novému hardvéru alebo softvéru, zmenám používateľských požiadaviek či používateľského rozhrania, prípadne kvôli zvládnutiu nárastu počtu používateľov a ich intenzity práce nebude veľmi náročné.

3.2 Spracovanie a vykresľovanie webových stránok

S rôznymi architektúrami sa spájajú rôzne spôsoby, ako používateľom poskytnúť želaný obsah. Medzi dnes najviac používanými prístupmi cez webový prehliadač sa najčastejšie rozlišuje vykresľovanie na strane serveru, vykresľovanie na strane klienta, hydratácia či statické generovanie stránok. Tieto postupy budú postupne popísané.

K postupom sú priložené ukážky komunikácie medzi serverom a klientom. Klient načíta stránku `/index.html`, odkiaľ sa následne presunie na stránku `/items.html`. Cieľom je demonštrovať prvé načítanie webu, potom nasledujúce prechody v rámci tohto webu a ako to prebieha prostredníctvom týchto metód. Prerušovaná čiara smerujúca nadol predstavuje plynutie času. Hrúbka šípky predstavuje objem dát prenášaných v smere šípky, jej sklon naznačuje dobu trvania prenosu a textový popis hovorí, aké dáta sa prenášajú. Výška blokov na strane klienta i servera naznačuje dobu vykonávania aktivity uvedenej vedľa nich. Na strane klienta je jednou vodorovnou čiarou vyznačený moment, kedy je stránka zobrazená, ale interakcie ešte nie sú možné. Dvojitá vodorovná čiara značí plné načítanie stránky, teda je aj interaktívna. Tvar predĺženého písmena „I“ ohraničuje úsek od odoslania požiadavky na načítanie stránky do jej plného načítania. Niektoré správy a operácie, ktoré majú vo všetkých spôsoboch takmer identický vplyv na vykresľovanie, sú vynechané.

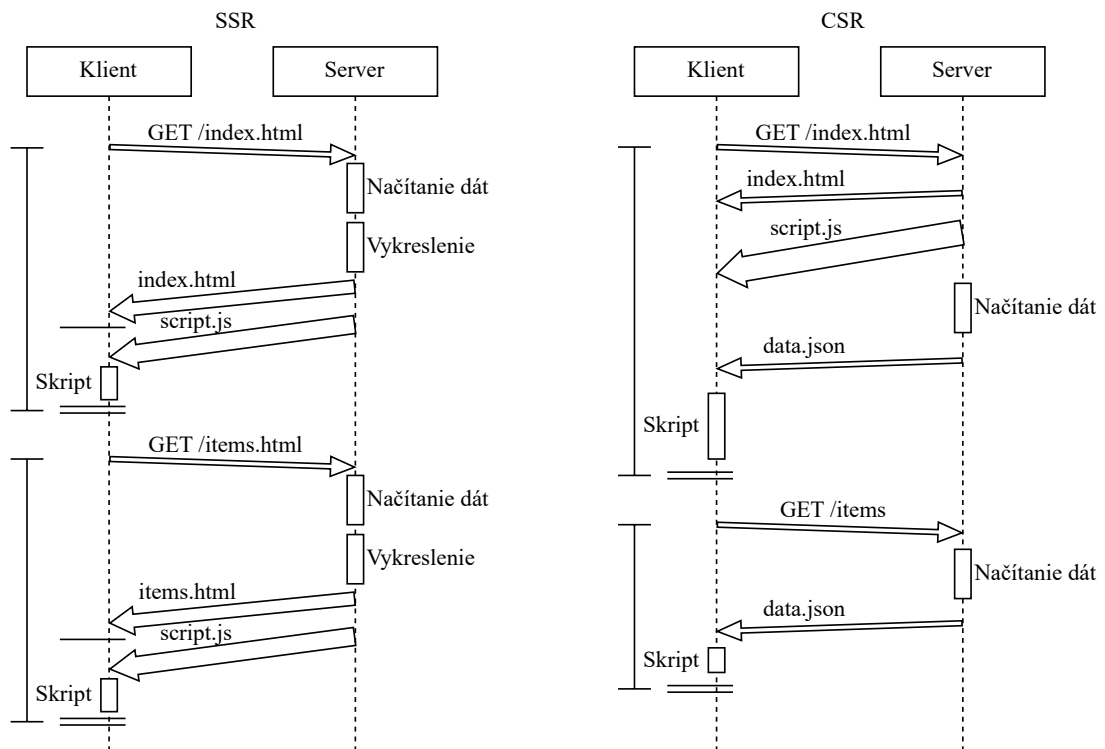
Vykresľovanie na strane serveru

V prípade, keď sa celý obsah stránky pripraví na serveri a následne sa pošle klientovi, ide o vykresľovanie na strane servera, anglicky *Server-Side Rendering (SSR)*. Klientský prehliadač si vyžiada stránku a server pripraví celý jej obsah v podobe HTML dokumentu, ktorý vráti klientovi. Klientský prehliadač aplikuje CSS pravidlá a zobrazí stránku. Medzitým prehliadač sťahuje JavaScript kód pre základnú interakciu s používateľom, po čom môže používateľ so stránkou plne interagovať. Tento spôsob umožňuje rýchlejšie načítanie stránok a lepšie SEO² výsledky pre vyhľadávače. Nevýhodou je vyššia záťaž serveru, nakoľko sa musí pri každej požiadavke zostaviť celá stránka, aj keď sa obsah mení len minimálne [30]. Obrázok 3.7 vľavo ukazuje načítanie dvoch stránok v podobe HTML dokumentov, ktoré vykreslil server.

Vykresľovanie na strane klienta

Vykresľovanie na strane klienta, anglicky *Client-Side Rendering (CSR)*, je realizované jazykom JavaScript v prehliadači klienta. Prehliadač najskôr načíta iba základný HTML dokument a potrebný JavaScript kód, podľa ktorého sa vykreslí zvyšok stránky. Kvôli potrebe preniesť väčší objem JavaScript kódu je prvotné načítanie stránky pomalšie, ale následné prechody medzi stránkami sú rýchlejšie, pretože sa zo serveru načítavajú iba žiadané dáta a nie celá stránka. Stránka je tvorená dynamicky a je plne dostupná až po vykonaní kódu na strane klienta. Presunutím vykresľovania na klienta sa znižuje záťaž servera, a tým sa rovnomernejšie rozkladá výpočtová náročnosť naprieč zariadeniami. CSR je vhodné pre aplikácie, ktoré poskytujú používateľom viac možností interaktivity a kde sa obsah často mení [31]. Na obrázku 3.7 vpravo je zobrazený iný princíp fungovania získava dát, napríklad z aplikačného rozhrania REST, ktoré bude popísané neskôr.

²SEO – skratka pre Search Engine Optimization, optimalizácia stránky pre vyhľadávače na zlepšenie návštevnosti



Obr. 3.7: Ukážka načítania dvoch stránok pri vykresľovaní na strane serveru (vľavo) a na strane klienta (vpravo).

Rehydratácia

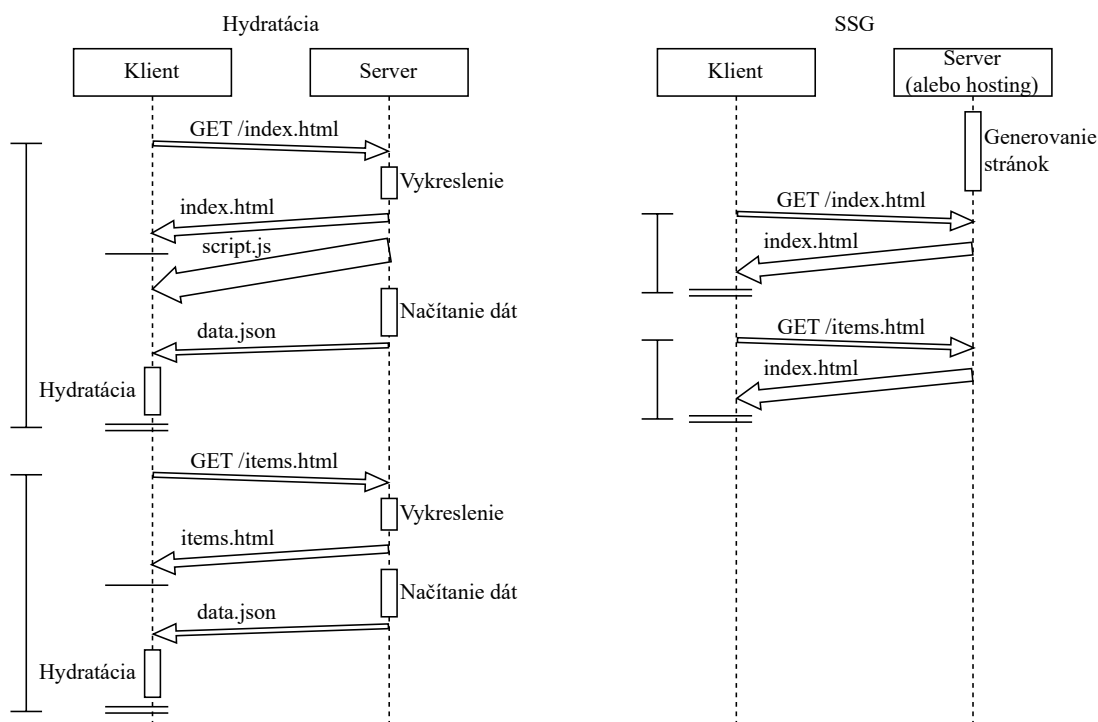
Rehydratácia, alebo aj zjednodušenie hydratácia, umožňuje webovej aplikácii dosiahnuť interaktívny stav po tom, čo je vykreslená na strane servera. Kombinuje teda výhody CSR aj SSR. Vďaka tomu je načítanie základu stránky rýchle a na klientovi sa dokreslí obsah žiadaný používateľom – stránka sa hydratuje. Hydratovaná stránka je plne interaktívna a používateľ s ňou môže pracovať [32].

Progresívna hydratácia je modifikácia, ktorá optimalizuje vykresľovanie a interaktivitu jednotlivých komponentov. Progresívna je v tom, že komponenty stránky sa vykresľujú postupne v čase podľa potreby a dôležitosti, čo skraca celkový čas načítania. Prípadné chýbajúce komponenty sa načítajú až v momente potreby. Stránka sa zobrazí, akonáhle prehliadač obdrží HTML dokument a začne sa doplňovať podľa načítaných dát [32].

Čiastočná hydratácia funguje na princípe takzvaných ostrovov (anglicky *islands*), kedy sa hydratujú iba vybrané komponenty bez potreby hydratovať všetky komponenty. Klientovi je doručený HTML dokument s obsahom vykresleným na serveri (časť SSR) a po načítaní JavaScript kódu sa upraví obsah stránky v rámci ostrovov (časť CSR). Takto sa prostredníctvom kódu JavaScript aktualizujú iba vybrané časti stránky a nie celá stránka, čo zvyšuje výkonnosť a efektivitu aplikácie [32]. Obrázok 3.8 vľavo naznačuje, ako prebieha hydratovanie stránok.

Statické generovanie stránok

Generovanie statických stránok, anglicky *Static Site Generation (SSG)*, je prístup, ktorý generuje podoby webových stránok vopred a klientovi sú poskytované v podobe statických HTML súborov. Vďaka tomu sú stránky načítané veľmi rýchlo a ich poskytovanie dokonca nevyžaduje žiadny server. Stránky sú taktiež bezpečnejšie, nakoľko sa nevykonáva žiadny kód na strane servera, ktorý by mohol obsahovať chyby. Vygenerované stránky sú úplne vytvorené, plne interaktívne, optimalizované a opakovane využívané v odpovediach na viaceré žiadosti. Tento postup je vhodný pre stránky, kde sa obsah mení iba zriedka [6]. Poskytovanie staticky vygenerovaných HTML stránok je znázornené na obrázku 3.8 vpravo.

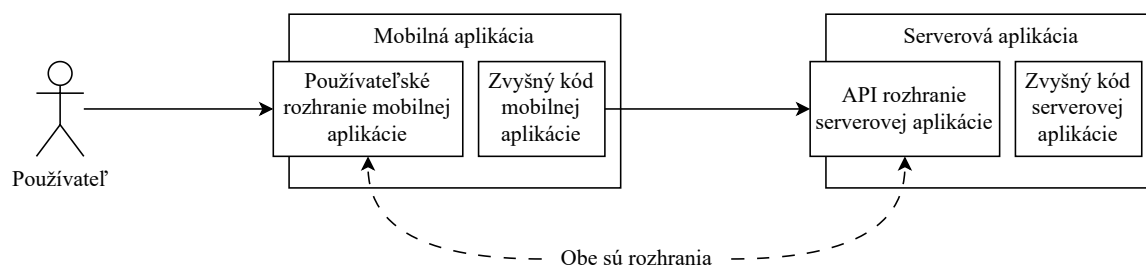


Obr. 3.8: Ukážka načítania dvoch stránok pri využití rehydratácie (vľavo) a načítanie dvoch staticky vygenerovaných stránok (vpravo).

3.3 Spôsob komunikácie aplikácií

Aplikácie, alebo ich časti, spolu komunikujú cez aplikačné programové rozhrania (*Application Programming Interface*, skrátene **API**). API je vo všeobecnosti rozhranie, teda bod, kde sa stretávajú a interagujú dva systémy, subjekty alebo organizácie. Takéto rozhrania môžu byť v kontexte informačných technológií poskytované pre ovládanie súčastí zariadení (napríklad fotoaparát mobilu), pre volanie funkcií pomocnej knižnice alebo pre vykonávanie služieb na vzdialených zariadeniach prostredníctvom internetu. Pre potreby tejto práce sa bude pracovať s webovým aplikačným rozhraním (web API), čo slúži ako webové rozhranie pre softvéry. Webové preto, lebo na komunikáciu sa využíva protokol HTTP, ktorý je základom pre komunikáciu po internete.

Obrázok 3.9 ilustruje analógiu používateľského rozhrania a aplikačného programového rozhrania na používaní mobilnej aplikácie, ktorá komunikuje so serverovou aplikáciou. Aplikačné rozhranie plní rovnakú úlohu voči inej aplikácii ako používateľské rozhranie voči používateľovi. Obe sú zodpovedné za riadenie interakcie medzi dvoma stranami. Zatiaľčo používateľské rozhranie využíva rôzne elementy pre interakciu s človekom, API na interakciu poskytuje funkcie, ktoré môžu prijímať dáta alebo ich vracaf. API je teda rozhranie softvéru, ktoré abstrahuje jeho vnútornú implementáciu. Ten, kto využíva rozhranie softvéru sa nazýva konzument (klient) a ten, kto ho poskytuje, teda aplikácia, sa nazýva poskytovateľ (server). Okrem aplikácií však s API rozhraním pracujú aj vývojári, pre ktorých je pri vývoji inej aplikácie dôležité nápomocné a jednoduché rozhranie. Aplikáciu, ktorá poskytuje API rozhranie, musel niekto vytvoriť, a tak toto API rozhranie je rozhraním aj pre ostatných vývojárov [18].



Obr. 3.9: Porovnanie používateľského rozhrania (UI) s aplikačným programovým rozhraním (API). (Inšpirované [18].)

Pri viacerých architektúrach v predchádzajúcej časti je spomenuté rozdelenie aplikácie na programy, prípadne programu na moduly, a ich vzájomná komunikácia. Ako môže prebiehať komunikácia medzi rôznymi časťami po internetovej sieti, bude ukázané na bežne používaných štýloch komunikácie – REST, GraphQL a gRPC.

Výber konkrétnej možnosti vždy vychádza z analýzy potrieb a kontextu použitia. Najskôr treba spoznať konzumenta rozhrania a jeho kontext; ciele, ktoré chce dosiahnuť; spôsoby, ako ich dosiahnuť a až potom možno vyberať najvhodnejšiu alternatívu. Zdrojmi informácií o spomínaných spôsoboch komunikácie sú [18, 9].

REST

REST je skratka pre *Representational State Transfer*. REST je architektonický štýl, ktorý predstavil Roy Fielding vo svojej dizertačnej práci [12] v roku 2000. Zámerom štýlu REST je uľahčiť tvorbu distribuovaných systémov, ktoré majú byť účinné, škálovateľné, spoľahlivé, ľahko znovu použiteľné, čitateľné a modifikovateľné. Akonáhle softvérová architektúra spĺňa všetky tieto vlastnosti (alebo sa o to snaží), možno ju označiť za „RESTful“. Na to musí architektúra vyhovovať všetkým nasledujúcim obmedzeniam (pomenovania obmedzení sú v angličtine pre zachovanie ich pôvodného významu):

- **Client/server separation** – V architektúre musí byť jasné oddelenie zodpovedností medzi komponentami klientskej aplikácie a jej API servera z hľadiska rozdelenia práce a vzájomnej komunikácie. Oddelené časti sa vyvíjajú samostatne a jediné, čo majú spoločné, je komunikačné rozhranie.

- **Statelessness** – Všetky informácie potrebné pre vykonanie jedného dotazu sú obsiahnuté v danom dotaze a na strane serveru sa medzi jednotlivými požiadavkami neukladajú žiadne informácie o klientovi. Každá požiadavka sa spracováva ako úplne nová a prípadná správa stavu je riešená na klientovi.
- **Cacheability** – Odpoveď na požiadavku musí uviesť, či ju možno uložiť a na ako dlho, pre účely opätovného použitia klientom namiesto odosielania rovnakej požiadavky.
- **Layered system** – Pri komunikácii klienta so serverom vidí tento klient iba jednu vrstvu systému a nie kompletnú infraštruktúru, ktorá sa skrýva za ňou.
- **Uniform interface** – Rozhranie musí jednoznačne identifikovať zdroje, ktoré sú použité v komunikácii medzi serverom a klientom. Klienti manipulujú so zdrojmi cez reprezentácie ich stavov, ktoré sú jednotné naprieč odpoveďami zo serveru. To znamená, že použitím unikátnych ciest zdrojov v spojení so štandardnými HTTP metódami sa dosiahne konzistentnosť v rámci jedného API, ale tiež s inými API rozhraniami štýlu REST. Interakcie tiež poskytujú všetky potrebné metadáta pre pochopenie reprezentácií a možných operácií so zdrojmi. Toto je základným obmedzením, nakoľko používanie REST rozhrania spočíva v prenose reprezentácií stavov zdrojov.
- **Code on demand** – Server môže poskytnúť klientovi spustiteľný kód (napríklad vo forme JavaScript kódu) na rozšírenie jeho funkcionalít; toto obmedzenie je voliteľné.

Hoci bol architektonický štýl REST pôvodne určený na popis protokolu HTTP, tieto princípy sa dajú uplatniť aj v iných architektúrach softvéru, ktoré majú obdobné potreby. API štýlu REST je založené na zdrojoch (*resource-based*) a využíva protokol HTTP, pričom svoje ciele reprezentuje pomocou akcií (HTTP metód) na týchto zdrojoch (URL cesty). HTTP má definované aj štandardné stavové kódy, ktoré rozhrania REST využívajú na informovanie klienta o výsledkoch operácií. Kódy sa delia do viacerých kategórií na odlíšenie ich využitia a každá kategória obsahuje kódy so špecifickým významom. Zakladanie si na zaužívanom protokole prispieva ku konzistencii a predvídateľnosti API rozhrania. Vďaka vlastnostiam protokolu HTTP, najmä ukládanie do vyrovnávacej pamäte a podmienené požiadavky, nemusí vývojár riešiť tieto záležitosti vo vlastnej réžii.

Použitie rozhrania REST je v dnešnej dobe najbežnejšia a z pohľadu rozhodovania najistejšia voľba. Vhodne navrhnuté rozhranie REST umožní vykonávanie väčšiny potrebných operácií. Je to zároveň najpoužívanejší spôsob tvorby API rozhrania a vďaka konzistentnosti naprieč API rozhraniami je jednoduché na pochopenie pre väčšinu vývojárov.

gRPC

gRPC je štandard pre volanie vzdialených procedúr, anglicky *Remote Procedure Call* od spoločnosti Google. Konzument rozhrania volá serverové procedúry, akoby boli jeho lokálne. gRPC je založené na systéme Stubby, ktorý Google používal v internej komunikácii vlastných mikroslužieb.

gRPC je tiež postavené nad protokolom HTTP, ale nakoľko ide iba o volanie funkcií, neposkytuje možnosť využívania vyrovnávacej pamäte. Naopak, vďaka využitiu HTTP 2 a formátu serializovaných dát *Protocol Buffers*³, ponúka vysokú výkonnosť, streamovanie a obojsmernú komunikáciu medzi konzumentom a poskytovateľom. Rozhranie gRPC založené

³Domovská stránka formátu Protocol Buffers: <https://protobuf.dev/>

na funkciách (*function-based*) nevyužíva štandardnú štruktúru akcií protokolu HTTP, a tak je na vývojárovi, aby určil vlastnú syntax a sémantiku takmer pre všetky operácie, či už je to získavanie dát alebo ich úprava, alebo formát reprezentácie dát. Chybové stavy v gRPC sú riešené štandardným modelom chýb tejto technológie, kde niektoré chybové kódy majú ekvivalentný význam ako tie z protokolu HTTP. Obsah chýb môže byť doplnený dodatočnými dátami podľa potreby konkrétnej aplikácie.

Použitie API rozhrania typu gRPC je vhodné, ak záleží na každej ušetrenej milisekunde v komunikácií medzi aplikáciami, ktoré sa v organizácii používajú interne. Ďalej je vhodné, ak sa rozhranie nebude zverejňovať a využitie vyrovnávacej pamäte nie je potrebné alebo sa dá nahradiť vlastnou implementáciou v rámci aplikácie.

GraphQL

GraphQL bolo vyvinuté spoločnosťou Facebook (dnes Meta) v roku 2012 a uvoľnené ako open-source program v roku 2015. Ide o dopytovací jazyk pre rozhrania API a behové prostredie pre vykonávanie týchto dotazov s existujúcimi dátami. Dôvodom vzniku tohto riešenia bola potreba aplikačného rozhrania dostatočne silného a zároveň jednoduchého na používanie. GraphQL poskytuje kompletný popis dát ponúkaných cez API a umožňuje klientovi žiadať iba o tie dáta, ktoré potrebuje [33].

Ak záleží na objeme prenášaných dát, každej milisekunde času alebo počte volaní vzdialených metód cez rozhrania API, možnosť výberu konkrétnych dát klientom a ničoho navyše je užitočnou vlastnosťou, ktorú GraphQL ponúka. Rovnako ako REST rozhrania, aj GraphQL rozhrania sú postavené na protokole HTTP, hoci dokážu pracovať aj na inom protokole fungujúcom na princípe žiadosť-odpoveď. Rozdielne sú v tom, že GraphQL využíva iba jednu HTTP metódu, a to POST, ktorú používa na všeobecnom zdroji s URL cestou `/graphql`. Nakoľko každá žiadosť je definovaná klientom v podobe objektu, ktorý môže mať rôznu veľkosť aj komplexnosť, telo metódy POST je na to vhodným miestom, kde sa nachádza vo forme reťazca pod položkou *query*. Obsahom GraphQL požiadavky je výpis dát, o ktoré klient žiada. Okrem toho GraphQL umožňuje odosielanie viacerých požiadaviek naraz, ktoré sú za sebou obsiahnuté v spomínanej položke *query*. Rovnakým spôsobom sú následne radené aj odpovede na zahrnuté požiadavky v HTTP odpovedi. Štruktúra požiadavky je veľmi podobná štruktúre dát v odpovedi, čo prispieva k jednoduchosti tvorby dotazov a spracovania odpovedí v očakávanom formáte. Nevýhodou využívania iba HTTP metódy POST môže byť nemožnosť ukladania odpovedí do vyrovnávacej pamäte, takže každá opakovaná žiadosť sa musí na serveri nanovo vyhodnotiť v celom rozsahu. To však neovplyvňuje prácu so stále sa meniacimi dátami, kde využívanie vyrovnávacej pamäte aj tak veľký význam nemá.

GraphQL je dopytovací jazyk pre rozhrania API (*API Query Language*), ktorý by sa dal prirovnať k jazyku SQL (*Structured Query Language*) využívaného pre dopytovanie sa nad databázami. Podobnosť vzniká už pri porovnaní ich názvov a dali by sa nájsť ďalšie podobné vlastnosti. Rozdielne sú však použitia v rámci architektúry softvéru. SQL sa využíva pre manipuláciu s dátami priamo v databáze, zatiaľčo GraphQL manipuluje s dátami na rozhraní medzi aplikáciami, teda cez ich aplikačné programové rozhrania.

Rozhranie GraphQL poskytuje konzumentovi schému popisujúcu ponúkané dáta, podľa ktorej si konzument vie vyžiadať presne tie dáta, ktoré potrebuje – operácie *queries*. Podľa toho možno označiť GraphQL ako rozhranie založené na dátach (*data-based*). V prípade inej operácie ako čítanie, napríklad vytváranie, úprava alebo mazanie záznamov, GraphQL sa správa rovnako ako RPC rozhranie, teda využíva funkcie, ktoré sa v tomto prípade

nazývajú *mutations*. GraphQL automaticky rieši chybové stavy tak, že odpoveď okrem (čiastočných) žiadaných dát obsahuje aj slovný popis chýb, ktoré nastali. Toto správanie môže byť doplnené podľa potreby konkrétnej aplikácie.

V porovnaní s REST API alebo RPC API, GraphQL neposkytuje kontrolu nad potenciálne vytváranými požiadavkami. Obsluhujúci server nemusí zvládať veľké množstvo komplexných požiadaviek, čo môže viesť k zahlteniu systému. Riešením by mohlo byť aplikovanie rôznych obmedzení na požiadavky, čím sa však znižuje uplatiteľnosť tohto prístupu. GraphQL je vhodné pre privátne API rozhrania so špecifickými potrebami, ktoré nedokážu byť vykonané rozhraním REST. Táto možnosť sa vyplatí aj v nasledujúcich situáciach: keď sú potrebné pokročilé dotazovacie možnosti; API rozhranie zostane privátne a nebude sa s nikým zdieľať; vyrovnávacia pamäť nie je dôležitá alebo keď systémy môže byť zabezpečený voči zahlteniu implementáciou či škálovateľnosťou [18].

Na porovnanie predstavených štýlov API rozhraní budú využité výsledky práce [9], kde autor T. Buňata vytvoril ekvivalentné API rozhrania spomínaných štýlov v prostredí Node.js⁴ a následne ich porovnával z viacerých hľadísk.

Z pohľadu množstva vybavených žiadostí za sekundu s miernym náskokom vedú rozhrania typu REST a GraphQL (dopytujúce iba časť dát); gRPC a GraphQL (tu dopytujúce všetky dáta) sú podľa autora pomalšie kvôli dlhšej serializácii a deserializácii správ v komunikácii. V porovnaní veľkosti odosielaných požiadaviek má návrh gRPC nasledované rozhraním REST. Oba prípady dotazov GraphQL sú výrazne objemnejšie, nakoľko klient žiada o dáta v konkrétnom tvare zapísanom v textovej podobe. Odpoveď cez gRPC je stále najmenšia, výber časti dát cez GraphQL je na druhom mieste, tretí skončil REST a najväčšiu odpoveď prenášalo GraphQL so všetkými dátami. Pri vytváraní nových záznamov sú REST a gRPC takmer rovnako rýchle, no GraphQL skokovo zaostáva. Poradie podľa priemernej veľkosti prenášaných správ pri vytváraní je na tom obdobne ako v prípade čítania dát.

Autorovi sa najrýchlejšie tvorilo rozhranie REST, implementáciu GraphQL označil za vývojársky prívetivú a gRPC bolo na začiatok najnáročnejšie. Staranie sa o dokumentáciu API je pri GraphQL a gRPC z princípu vynútené technológiou; dokumentácia pre REST zostáva v rukách vývojárov.

3.4 Existujúce nástroje pre podporu procesného riadenia

Nakoľko sa informačné systémy stali každodennou súčasťou podnikov, vzniklo mnoho podôb aplikácií so špecifickým zameraním na určitú oblasť fungovania podniku. Medzi ne patria aj softvéry Camunda a OpenProject.

Camunda

Nástroj Camunda od rovnomennej spoločnosti predstavuje platformu pre organizáciu podnikových procesov definovaných v jazyku BPMN verzie 2.0. Jadro tejto platformy je napísané v jazyku Java a viaceré moduly sú dostupné v podobe open-source softvéru, no v ponuke je tiež platená verzia. Táto platforma poskytuje celú škálu prostriedkov pre prácu s procesmi, ktoré sú rozdelené do viacerých modulov a ich popis pochádza z [10].

⁴Node.js je behové prostredie pre aplikácie napísané v jazyku JavaScript, viac na <https://nodejs.org/>

Modeler slúži na návrh procesov v jazyku BPMN a rozhodovacej logiky v jazyku DMN⁵. Vďaka využitiu týchto vizuálnych jazykov sa uľahčí pochopenie a zlepši spolupráca medzi technickými a netechnickými používateľmi. Tento modul je dostupný aj ako samostatná aplikácia, v ktorej sa dajú definovať procesy v jazyku BPMN a následne uložiť v podobe XML dokumentu. Pri použití v rámci platformy Camunda je možné simulovať behy procesov a tak už pri ich definícii odstrániť niektoré chyby, ktoré by sa inak objavili až v produkčnom nasadení. Rozhodovacia logika má podobu rozhodovacej tabuľky, kde sa kombináciám vstupných hodnôt priradí výstupná hodnota.

Modul *Forms* ponúka možnosť tvorby formulárov spôsobom „drag & drop“ z určenej množiny formulárových polí. Tieto formuláre môžu byť pridružené k prvkom diagramu BPMN a v rámci platformy Camunda priamo poskytované používateľom, keď sa proces dostane do stavu reprezentovaného daným prvkom. Tento modul sa taktiež dá využívať ako samostatná aplikácia.

Workflow Engine je behové prostredie pre vykonávanie procesov workflow podľa ich BPMN zápisu. Poskytuje automatizované spracovávanie a vykonávanie podnikových procesov na základe využitých elementov jazyka BPMN. *Decision Engine* je obdoba pre vykonávanie automatizovaného rozhodovania, ktoré nastáva v bránach BPMN diagramu pre určenie nasledujúceho smeru toku procesu.

Nástroj *Optimize* umožňuje v podobe prehľadných dashboard obrazoviek analyzovať behové štatistiky procesov, napríklad výkonnosť, čas strávený v jednotlivých aktivitách, či dokonca detegovať nevyužívané cesty. Na základe nich možno optimalizovať procesy pre efektívnejšie vykonávanie.

S platformou Camunda sa dá komunikovať aj prostredníctvom API rozhrania štýlu REST. Toto rozhranie je popísané špecifikáciou OpenAPI, čo uľahčuje písanie kódu pre komunikáciu medzi programami.

Špecifikácia OpenAPI je formálny štandard pre popis rozhraní komunikujúcich cez protokol HTTP. Je nezávislý na programovacom jazyku a uľahčuje pochopenie možností služieb bez potreby skúmania zdrojových kódov, či analýzy sieťovej komunikácie [20].

OpenProject

OpenProject je open-source softvér pre riadenie projektov v klasickej alebo agilnej podobe, ktorý poskytuje podporu pre všetky fázy životných cyklov projektov. Z agilných metodík podporuje Kanban a Scrum. Softvér je napísaný prevažne v jazyku Ruby a komunitná verzia je voľne dostupná, ale na využitie všetkých rozširujúcich podnikových funkcionalít treba zvoliť jeden z platených variantov [22].

V rámci tohoto softvéru sa dajú riadiť celé portfólia projektov z jedného miesta, kde každý môže byť riadený inou metódou. Projekty sa môžu plánovať, určovať ich prioritu a monitorovať priebeh ich vykonávania pre optimálne využitie zdrojov s obmedzenou dostupnosťou. V projektoch možno vytvárať ako úlohy, tak aj podprojekty, ktoré sa hierarchicky usporadúvajú do stromovej štruktúry. Pre zvýšenie prehľadnosti časových nadväzností je dostupné zobrazenie v podobe Gantt diagramov. Projekt na najvyššej úrovni sa dá chápať aj ako organizačná jednotka v rámci spoločnosti, takže podprojekty v systéme by, napríklad, predstavovali projekty rozdelené podľa oddelení. Projekty môžu mať stanovené mílniky a taktiež môžu byť rozdelené na viacero fáz pre lepšie definovanie čiastkových cieľov. Všetky dôležité informácie o projekte sú združené na dashboard obrazovke, ktorá je plne konfigurovateľná. Obsahuje napríklad popis projektu, zoznam ľudí priradených na pro-

⁵DMN – skratka pre Decision Model and Notation, viac na <https://www.omg.org/dmn/>

jekt, uskutočnené zmeny alebo prehľad rozpracovaných a uzavretých úloh. Každá zmena je zaznamenaná a ich následnosť je dostupná v chronologickom zobrazení. Pre projekty sa dajú vytvárať wiki stránky, ktoré na jednom mieste združujú všetky potrebné informácie o projekte, napríklad projektovú dokumentáciu, referencie, inštrukcie alebo manuály.

Pre opakujúce sa projekty, ktoré vyžadujú určitú štruktúru, umožňuje OpenProject definovať šablóny. Vďaka tomu sa urýchljuje zakladanie nových projektov a udržiava sa konzistencia. Užitočná je aj možnosť vytvoriť šablónu z už existujúceho projektu, takže ak vyhovuje štruktúra nejakého skúšobného projektu, dá sa z neho spätne vytvoriť šablóna pre ďalšie projekty.

Domovská stránka každého používateľa je tiež realizovaná formou konfigurovateľnej dashboard obrazovky, ktorá, v tomto prípade, ponúka zoznam pridelených úloh používateľovi vo všetkých projektoch, hlásenia z projektov či kalendár s udalosťami a termínmi. Používateľ si taktiež môže definovať, o aké notifikácie má prednostný záujem a pri zmene stavu projektu alebo úlohy mu príde oznam do notifikačného centra. Notifikácia sa zašle aj v prípade, keď daného používateľa označí (znakom @) iný v komentári k aktivite alebo mu priradí úlohu. Okrem toho môžu používatelia po dokončení úlohy uviesť, koľko času potrebovali na vypracovanie úlohy, čo poslúži na analýzu nákladov na projekt či plánovanie budúcej práce.

Aj nástroj OpenProject ponúka na komunikáciu s inými aplikáciami rozhranie štýlu REST, prostredníctvom ktorého možno pracovať s rozsiahlym množstvom zdrojov v inštancii tohto systému, vrátane vytvárania nových úloh či získanie stavu vybraného projektu. Toto rozhranie je tiež špecifikované vo formáte OpenAPI.

Kapitola 4

Analýza súčasného stavu a požiadaviek firmy EKOM

Spoločnosť EKOM spol. s r.o., sídliaca v Piešťanoch, Slovenská republika (ďalej iba ako **EKOM**), sa zaoberá výrobou špecializovaných bezolejových kompresorov a odsávacích zariadení, ktoré sú určené na použitie v dentálnych, medicínálnych, laboratórnych i priemyselných aplikáciách. Spoločnosť EKOM sa interne delí na dve divízie, a to „Kompresory“ (skrátene KOM) a „Lisovňa“ (skrátene CNC). Nakoľko väčšina výrobkov sú zdravotnícke zariadenia, spoločnosť EKOM musí dodržiavať prísne normy, najmä ISO 9001 – Systém riadenia kvality a ISO 13485 – Zdravotnícke prostriedky, ktoré, okrem iného, určujú spôsoby narábania a spracovania informácií. Tieto normy taktiež stanovujú, že organizácia musí vytvoriť a udržiavať záznamy potrebné na preukázanie zhody s normou pre rôzne postupy, napríklad pre riadenie zmien, servisné činnosti, vybavovanie sťažností a podobne.

Pre účely dodržiavania týchto noriem sa v spoločnosti EKOM zaviedli procesy, ktorých úlohou je spracovávať informácie v súlade s uvedenými normami. Aktuálne sa v spoločnosti EKOM používajú nasledovné procesy:

- **Návrh na zmenu** – zbiera informácie o výsledkoch preskúmania zmien, zaznamenáva osobu, ktorá rozhodla o zmene a o činnostiach, ktoré vyplynuli z tohto preskúmania.
- **Zákazková úloha** – udržiava záznamy o výsledkoch preskúmania požiadaviek zákazníkov a činnosti vyplývajúce z nich.
- **G8D report** – uchováva záznamy o zaobchádzaní so sťažnosťami, záznamy o výsledkoch akéhokoľvek šetrenia a o prijatých činnostiach, či už ide o nápravné alebo preventívne činnosti.
- **Reklamácia dodávateľská** – rieši záznamy s povahou nezhôd a prijaté opatrenia voči nim.
- **Spätná väzba** – zbiera informácie o naplňaní požiadaviek zákazníkov a dokumentuje metódy na získavanie a používanie týchto informácií.
- **Servisné záznamy** – obsahuje záznamy o servisných činnostiach vykonávaných samotnou firmou alebo jej dodávateľom.

V spoločnosti EKOM sa od svojho vzniku v roku 1992 vystriedalo viacero foriem práce s týmito informáciami. V začiatkoch sa spracovávali ručne na papieri. Hromadenie papierov,

nejednotný formát a niekedy obtiažna čitateľnosť si vyžadovali zmenu. Zlepšením dostupnosti výpočtovej techniky a spopularizovaním kancelárskych softvérov sa začalo pracovať s informáciami prostredníctvom tabuľkových procesorov, najmä program Microsoft Excel. Ani to však nebolo dostatočné. Bol potrebný centrálny systém pre správu viacerých typov dokumentov s evidenciou toku informácií, vyjadreniami kompetentných ľudí a taktiež rozhodnutiami k danému predmetu. Keďže neexistoval softvér, ktorý by takéto požiadavky naplňal, spoločnosť EKOM si musela vytvoriť vlastný. Aktuálne používaný systém v priebehu rokov už zastaral, stal sa ťažkopádnym a akákoľvek zmena v ňom vytvára iba obavy o celkovú stabilitu systému.

Vedenie spoločnosti EKOM teda zvolilo cestu vývoja nového systému, ktorý by mal byť postavený na moderných technológiách, jednoduchý na používanie, rýchlejší a dynamickejší. Uprednostňovaný je jazyk TypeScript¹, v ktorom sú tvorené všetky nové aplikácie používané vo firme. Nakoľko sa jedná o systém, ktorý spracúva každodenné operácie, pôjde o informačný systém prevádzkovej úrovne (TPS) s paralelným prístupom pre viacerých zamestnancov firmy. Nový systém má byť schopný sa rýchlejšie adaptovať na úpravy v procesoch výroby a lepšie reagovať na legislatívne zmeny. Okrem toho má umožniť prácu so zložitejšími procesmi, čo doteraz nebolo možné, nakoľko všetky aktuálne procesy sú vnútornou implementáciou rovnaké a líšia sa iba v drobnostiach. Ďalej firma požaduje formálne definovanie procesov prostredníctvom štandardizovaných formátov, ideálne v jazyku BPMN. Doterajšie procesy neboli definované žiadnym spôsobom. Procesy sa bežne v priebehu času upravujú a firma požaduje sledovanie týchto zmien pre určovanie, podľa akej verzie proces prebiehal. Používateľské rozhranie má podporovať zobrazenie minimálne v slovenčine a v angličtine. Predpokladá sa integrácia overovania identity používateľa s existujúcou štruktúrou Active Directory, vďaka ktorej sa používatelia budú prihlasovať jednotnými prihlasovacími údajmi do viacerých firemných aplikácií.

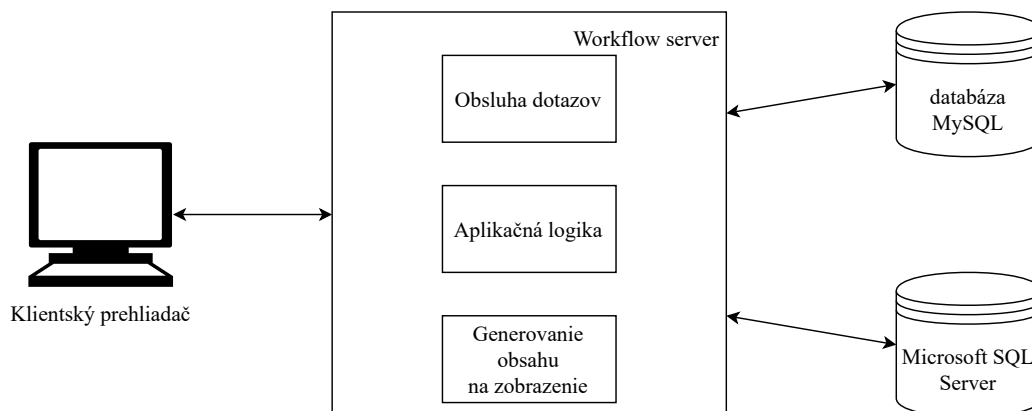
Aktuálne využívaný systém v spoločnosti EKOM má názov „Workflow“ a do prevádzky bol zavedený v roku 2008. Je napísaný v programovacom jazyku PHP verzie 5.1 a na ukladanie dát využíva databázu MySQL. Systém beží prostredníctvom servera IIS manager² ako interná webová aplikácia v rámci firmy a nie je prístupná cez internet, ale iba z lokálnej siete vnútri firmy. Databázový server, ako aj server so službou IIS manager, kde beží aplikácia Workflow, sa teda fyzicky nachádzajú vo firme. Približná architektúra systému je zobrazená na obrázku 4.1. Ako bolo naznačené, databáza MySQL sa nachádza na osobitnom databázovom serveri. Systém Workflow má prístup na čítanie aj z ďalšej firemnej databázy, tentokrát Microsoft SQL Server, ktorá sa využíva softvérom QAD³ určenom na riadenie, plánovanie a evidenciu výroby. Odtiaľ server Workflow získava artiklové čísla všetkých dielov pri kontrole vstupných polí formulárov, ktoré vyžadujú zadanie artiklového čísla. Od začiatku roku 2024 prechádza firma na riadenie výroby softvérovým riešením od spoločnosti SAP⁴, ktoré má nahradiť zdroj artiklových čísel pre nový informačný systém. Samotný systém Workflow beží na virtuálnom zariadení a rieši spracovávanie dotazov, vykonávanie aplikačnej logiky, správu používateľov, ako aj vytvorenie zobrazovaného obsahu, ktorý sa vracia používateľovi do prehliadača. Medzi používateľom a systémom Workflow sa nenachádza žiadna vrstva, ktorá by bola zameraná na toto rozhranie. Používatelia teda priamo pristupujú k súborom na serveri, ktoré implementačným kódom generujú HTML obsah.

¹Domovská stránka jazyka TypeScript: <https://www.typescriptlang.org/>

²IIS – Internet Information Services, viac na <https://www.iis.net/>

³Softvér QAD od spoločnosti Minerva, viac informácií na stránkach spoločnosti: <https://www.minerva-is.eu/sk/>

⁴SAP je producent podnikových aplikačných softvérov, viac na <https://www.sap.com/>



Obr. 4.1: Schéma aktuálne používaného systému Workflow v spoločnosti EKOM.

V roku 2018 prebehla revitalizácia systému Workflow na verziu 2. Táto obnova spočívala predovšetkým v aktualizácii PHP na verziu 7.2.9, pričom už vtedy bol systém technologicky aj ideologicky zastaralý bez možnosti dohľadať niektoré závislosti. Ďalšou aktualizáciou PHP, čo i len na verziu 7.3 sa systém stal úplne nefunkčným a jediným východiskom pre zachovanie funkčnosti bol návrat na verziu 7.2.9. Verzií 7.2 skončila podpora 30. novembra 2020⁵.

Z funkcionálneho hľadiska bol však významnejší rok 2014. Namiesto vlastného formátu reklamácií sa zaviedlo používanie techniky 8D-Report⁶. V porovnaní s ostatnými procesmi sa procesy techniky 8D-Report líšia v tom, že sa skladajú z ôsmich sekcií, kde každá sekcia rieši problém z iného uhla pohľadu. Okrem toho disponujú vyššou flexibilitou, nakoľko tok dát môže byť ovplyvnený v priebehu procesu podľa rôznych vstupov a nemusí sa nutne prechádzať všetkými sekciami. Sekcie sa skladajú z vlastných formulárových komponentov, ktoré zbierajú informácie dôležité pre danú sekciu, respektíve z daného uhla pohľadu. Nakoľko sa zvolila nevhodná granularita týchto komponentov, značná časť ich implementácie vznikla štýlom „skopíruj a vlož“, čo viedlo k obtiažnej starostlivosti o kód.

Používateľov spravuje administrátor v rámci systému. Tí sú pevne priradení do vopred vytvorených skupín pre účely tohto systému a tak nijako neodzrkadľujú organizačnú štruktúru vo firme. Prístupové práva sú dané individuálne pre každý modul formou ACL⁷ bez možnosti dodatočnej konfigurácie.

Používateľské rozhranie webovej aplikácie je vytvorené pomocou šablónového systému Smarty⁸ pre jazyk PHP a obsahuje minimum vlastného CSS a JavaScript kódu. Na interaktivitu medzi webom a používateľom sa využíva najmä jQuery. Domovská stránka webovej aplikácie je zobrazená na obrázku 4.2. Vo vrchnej časti je navigácia medzi modulmi, tlačidlo na vytvorenie nového dokumentu, tlačidlo na prístup ku konfigurácii systému a používateľské možnosti, napríklad odhlásenie sa zo systému. Pod navigáciou sa nachádza obsah špecifický pre jednotlivé moduly.

Systém Workflow sa skladá z nasledujúcich modulov. Modul *Dokumenty* slúži pre riadenie procesov formou dokumentov, ktoré sa vykonávajú vopred pevne danými krokmi.

⁵Oficiálne informácie o ukončení podpory verzií jazyka PHP: <https://www.php.net/eol.php>

⁶G8D – Global Eight Disciplines, analytická technika skladajúca sa z ôsmich častí pre riešenie neočakávaného problému vyvinutá spoločnosťou Ford

⁷ACL – access control list, zoznam pre riadenie prístupu

⁸Smarty template engine: <https://www.smarty.net/>

Stav	Dokument	Názov	Podal	Vyj.	Typ dokumentu	Dátum prid.	Dátum rozh.	Zav. do praxe	Zobraz
7	O_4136-2023	Back Shield - 714661005-000		1/3	Odchýlky CNC	1.12.2023			
2	Z_4135-2023	Návrh zmeny poisťovania záv. spojov kov+plast		0/8	Zmeny KOM	1.12.2023			
2	Z_4134-2023	Návrh na zmenu Základných vybavení		1/5	Zmeny KOM	1.12.2023			
2	Z_4133-2023	Návrh zmeny v el. zapoj. schemach.		1/5	Zmeny KOM	1.12.2023			
2	Z_4132-2023	Korekcia KP, stitku a NP		1/5	Zmeny KOM	1.12.2023			
2	..._GBDI_184-2023	EVERYDENT - netesni prepojenie TANDEM 2		S1_D...	DI G8D report	30.11.2023			
3	..._GBDI_183-2023	EUR-MED- poskodena elektronika		S3_D4	DI G8D report	30.11.2023			
2	Z_4131-2023	Technologická zmena XXI		4/5	Technologické Zmeny	29.11.2023			
2	ZUK_33-2023	Kompresor M40-110 VDC DRY pre SBB FV-Dosto - ALSTOM		10/15	Zákazková úloha-KOM	29.11.2023			
2	RKD_128-2023	Staff - spojka rozoberateľna		1/3	Reklamácia KOM dodav	29.11.2023			
2	RKD_127-2023	Condor - pressure switches		2/3	Reklamácia KOM dodav	29.11.2023			
2	RKD_126-2023	OEZ - stykače		2/3	Reklamácia KOM dodav	28.11.2023			
2	RKD_125-2023	SMC - snímač prietoku		1/3	Reklamácia KOM dodav	28.11.2023			
2	Z_4130-2023	Aktualizácia dokumentácie pre SADA NA PREROBENIE 2V/110M, ...		4/6	Zmeny KOM	27.11.2023			
2	..._CBK_16-2023	EKOM - AIR - poskodena elektronika		D4	KOM G8D report	27.11.2023			
3	..._GBDI_182-2023	Ducray Lenoir Ltd - poskodena elektronika		S3_D4	DI G8D report	27.11.2023			
3	RCD_27-2023	MAG45 nesprávna svorkovnica 024003370-000		2/2	Reklamácia CNC dodav	27.11.2023			
3	..._GBDI_181-2023	Halmađent- poskodeny susic		S3_D4	DI G8D report	27.11.2023			
2	RKD_124-2023	Cattani - držiak s hadicami		2/3	Reklamácia KOM dodav	24.11.2023			
3	RKD_123-2023	Danfoss - snímač teplotný		3/3	Reklamácia KOM dodav	24.11.2023			

Obr. 4.2: Domovská obrazovka systému WorkFlow používaného v spoločnosti EKOM.

Ukážka zoznamu dokumentov je na obrázku 4.2. Úlohou modulu *Projekty* je manažment projektov týkajúcich sa hlavne vývoja výrobkov. Výpis projektov vedených v systéme WorkFlow je zobrazený na obrázku 4.9. Modul *Úlohy* sa zameriava na sledovanie a vykonávanie úloh vyplývajúcich zo záverov jednotlivých dokumentov alebo patriacich pod niektorý z projektov. Príklad obsahu tohto modulu je na obrázku 4.11. Posledným modulom sú *Štatistiky*, ktorý možno využiť na spracovanie údajov z dokumentov do formy tabuliek alebo exportovanie požadovaných dát do súboru vo formáte XLSX. Ponuka modulu je na obrázku A.1.

Modul „Dokumenty“ je hlavnou časťou systému WorkFlow, kde sa vykonávajú podnikové procesy vyplývajúce z požiadaviek noriem ISO 9001 a ISO 13485. Zoznam dokumentov je zobrazený na obrázku 4.2. Modul má záložky:

- *Dokumenty* – zoznam dokumentov s možnosťou vyhľadávania.
- *Schválené* – zoznam schválených dokumentov prihláseného používateľa.
- *Zamietnuté* – zoznam zamietnutých dokumentov prihláseného používateľa.
- *Návrhy* – zoznam dokumentov, ku ktorým je prihlásený používateľ priradený na prvotné vyjadrenie a ešte tak nevykonan.
- *Rozhodnutie* – zoznam dokumentov čakajúcich na rozhodnutie.

Pod týmito záložkami sa vždy nachádza tabuľka s dokumentami, v ktorej možno filtrovať nad vybranými dátami. V záložke *Dokumenty* sú to: označenie dokumentu, názov dokumentu, meno podávateľa, stav, typ a G8D rozhodnutia. Stĺpce v tabuľke obsahujú informácie o dokumente v danom riadku. Význam stĺpcov je nasledovný (nevymenované stĺpce majú jednoznačný význam):

- *Stav* – číselné označenie aktuálneho stavu dokumentu, ku ktorému je priradená farba podľa legendy v spodnej časti. Toto číslo zodpovedá sekciam procesom G8D a späťne sa pridali aj k ostatným typom.
- *Dokument* – označenie dokumentu skladajúce sa zo skratky typu, automaticky zvyšovaného počítadla a roku podania.

- *Vyj.* – skratka pre vyjadrenia; udáva počet prijatých vyjadrení v pomere k počtu vyžiadaných vyjadrení.
- *Dátum prid.* – skratka pre dátum pridania; dátum vytvorenia dokumentu.
- *Dátum rozh.* – skratka pre dátum rozhodnutia; dátum finálneho rozhodnutia o prijatí alebo zamietnutí obsahu dokumentu.
- *Zav. do praxe* – skratka pre zavedené do praxe; dátum zavedenia obsahu dokumentu do praxe. Spája sa s výrobným číslom prvého výrobku, pri ktorého výrobe sa aplikuje tento dokument.

Príklad detailu jednoduchého dokumentu je na obrázku 4.3. Sekcia *Dokument* obsahuje dáta z vyplneného úvodného formulára podľa konkrétneho typu dokumentu a voliteľné prílohy. Nasleduje sekcia *Vyjadrenia*. Tu sa zbierajú vyjadrenia zvolených ľudí a ich stanovisko k predmetu dokumentu. Vyjadrenia môžu prebiehať vo viacerých kolách. Sekcia *Rozhodnutie* obsahuje finálne rozhodnutie zodpovednej osoby o prijatí alebo zamietnutí dokumentu. Úplne dolu sa nachádzajú úlohy pridružené k tomuto dokumentu a možnosť zapísania detailov o zavedení dokumentu do praxe.

Obr. 4.3: Príklad jednoduchého procesu vo forme dokumentu.

Zložitejší dokument G8D je zobrazený na obrázku 4.4. V tejto ukážke sú viditeľné časti 1 (D1 Vytvorenie tímu), 2 (D2 Popis problému), 3 (D3 Prijatie dočasných opatrení), 4 (D4 Analýza kľúčových príčin) a 8 (D8 Rozhodnutie). Ako bolo spomínané vyššie, každá časť hodnotí predmetnú udalosť z iného uhla pohľadu a zbiera iné dáta. Zber prebieha formou formulárov, ktoré sú individuálne definované pre každú časť. Dokument G8D nemusí nutne prechádzať cez všetky časti. V dokumente v ukážke sa prechádza z časti 4 rovno na časť 8. Časti 5, 6 a 7 sa vynechali. Za samotným dokumentom G8D sa nachádza zoznam príloh zozbieraných v priebehu procesu.

Šablóny pre dokumenty konfiguruje administrátor vo vizuálnom nástroji, ktorý je súčasťou systému (obrázok 4.5). V tomto nástroji sa nachádzajú zoznamy dostupných šablón, číselníkov a predpripravených tabuliek. *Šablóny* definujú podobu úvodných formulárov a je

D1 Vytvorenie tímu						
Člen	_		Člen	_		
Notifikovať členov tímu						
D2 Popis problému						
Poruchy mechanické prvky			Trend príčiny	Artikel	Zlyhaný diel	Množ.
				024000881-000	LOZISKO ZABEHNUTE, 620	1
D3 Prijatie dočasných opatrení						
diel zaslaný dňa: 21.06.2022			zákazka: DKR22149			
dodací list č.: KDL-1446-22			prepravné náklady: 0 €			
artikel:	názov:	množstvo:	fakturované:	výrobné číslo/sarža:	cena:	
024000881-X00	Lozisko	1	Nie		0	
604021396-X00	Piest s ojnicou	1	Nie		0	
					náklady na materiál spolu: 0 €	
Dočasné opatrenia:						
číslo	názov úlohy	dátum	riešiteľ	priebeh	kontroloval	dátum
Informácie o dotknutých osobách						
Ohrozil ZP zdravie užívateľa/obsluhu/inej zainteresovanej osoby			Nie	Bol pacient postihnutý?		
C: Udalje o pacientovi (vyplňa sa len v prípade keď bol pacient pripojený)						
Posúdenie prípadu						
incident:	priorita 0	Incident (Udalosť a problém výroby) nespôsobil ani by nemohol spôsobiť alebo prispieť k smrti alebo vážnemu zhoršeniu zdravia alebo ZP neprispel k tomuto incidentu.				
		Reakcia – Incident nie je potrebné hlásiť.				
opatrenie:	priorita A	Cimnosti zavedené v organizácii sú dostatočné na zabezpečenie výkony a kvality ZP, podporené negatívnym trendom opakovaného výskytu daného incidentu.				
		Reakcia - Opatrenie nie je potrebné vykonať, sťažnosť sa uzatvorá s dočasným nápravným opatrením.				
Rozhodol:						
D4 Analýza kľúčových príčin Ishikawa						
Vyjadrenie:						
Analýza:						
číslo	názov úlohy	dátum	riešiteľ	priebeh	kontroloval	dátum
	Žiadne úlohy					
D8 Rozhodnutie						
Akceptované: Nie						
Rozhodnutie:						

Vybrať súbory

(Maximálna veľkosť jednej prílohy: 150 MB)

2022-06-15.claim-email.pdf

Obr. 4.4: Ukážka sekcií G8D dokumentu.

možné zakázať tvorbu nových dokumentov podľa danej šablóny alebo sa prepnúť na úpravu šablóny. Sekcia *Číselníky* zobrazuje aktuálne stavy používaných číselníkov pre označovanie jednotlivých dokumentov. *Tabuľky* obsahujú vstupné pole s pripravenou tabuľkou a pripravenými stĺpcami.

V rozhraní na obrázku 4.6 môže administrátor nastaviť názov a popis šablóny, priradiť skupinu schvaľovateľov a určiť počítadlo pre číslovanie dokumentov. Hodnota tohoto počítadla sa používa pre označenie dokumentu, napríklad ZUK_1234-2023 označuje v poradí 1234. zákazkovú úlohu pre divíziu Kompresory v roku 2023. Toto počítadlo sa automaticky zvyšuje o 1 s každým novým dokumentom daného typu. V niektorých prípadoch sa s novým rokom nastaví na nulu, inokedy pokračuje s nastavenou hodnotou. Zaujímavosťou je, že všetky typy dokumentov majú vlastné počítadlo s výnimkou zmenových konaní a odchylok, ktoré sa číslujú podľa spoločného počítadla. Zámerom bolo ľahšie sledovanie vzájomnej nadväznosti týchto dvoch typov. Formuláre procesov sa skladajú z vytvorených

Dostupné šablony			
Názov	Popis	Upraviť	Pristup
Zmeny KOM	šablona zmeny KOM	Upraviť	Zakázať
Odchylky CNC	šablona odchylky CNC	Upraviť	Zakázať
Zakazkový list KOM	šablona Zakazkovy list KOM	Upraviť	Zakázať
Zakazková uloha CNC	šablona zakazkovej ulohy CNC	Upraviť	Zakázať

Vytvorit nový template | Vytvorit nový template r8

Číselníky		
Aktuálny stav	Popis	Upraviť
Z_4137-2023	zmenove konania	Upraviť
ZUK_34-2023		Upraviť
G8D_1-2023	8D report	Upraviť
O_4137-2023 <i>prepojený s Z_4137-2023</i>		Upraviť


Vytvorit nový číselník

Tabuľky (komponenta)	
Názov	Upraviť
Reklamácie KOM-material	Upraviť
Reklamácia KOM-dodavateľ	Upraviť
Slaznosť -medical	Upraviť
8D- nudzove opatrenie	Upraviť

Vytvorit novú tabuľku

Obr. 4.5: Zoznam položiek na konfiguráciu.

komponentov, teda vstupných polí pre formuláre, z ktorých sa už mnoho nepoužíva alebo sú vytvorené duplicitne. Pre komponenty sa v časti *konfigurácia komponentu* nastavujú vizuálne vlastnosti, podľa ktorých sa zobrazujú vo formulároch. Vstupy formulárov sa nedajú označiť ako „povinné“ a tak sa bežne zabúda na vyplnenie dôležitých polí. Chýbajúce údaje sa musia ručne dopísať v databáze správcom systému.

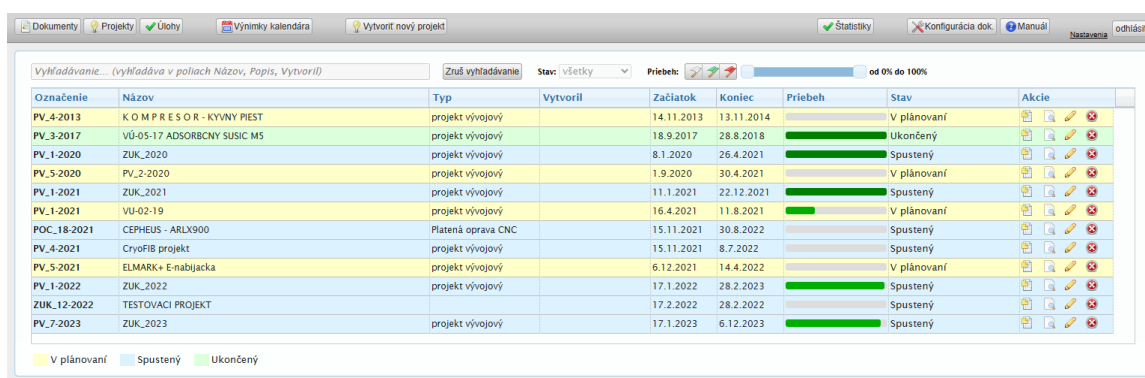
Nastavenia šablony.	Návrh formulára
Meno šablony: Zmeny KOM Schvaľovateľ: Z- Schvaľovateľ KOM Popis šablony: šablona zmeny KOM Číselník: Z_4137-2023 Chronologické zobrazovanie: <input type="checkbox"/> Špeciálny WF: <input type="checkbox"/> Zrýchlené vyjadrovanie: <input type="checkbox"/>	presunanim okien editujete vzhľad formulára  NÁVRH NA ZMENU Číslo dok.: XX_XXX_2008 Ekom spol s r.o. Podaná dňa: 04.12.2023 Dokument podal: Meno dokumentu: Popis: Je zmena podstatná/významná?:
Pridanie komponentu <ul style="list-style-type: none"> Linkovanie WF Tabuľka Nápis Selectbox Checkbox 	

Obr. 4.6: Konfigurácia formuláru jednoduchého procesu.

Ukážka časti komplexnejšej konfigurácie procesu G8D je na obrázku 4.7. Tu sa editujú jednak sekcie dokumentu G8D a tiež použité komponenty v sekciách. Sekciám sa dá prideliť farba pre ľahšie vizuálne oddelenie. Zmeny vlastností dokumentov, zloženia formulárov, ako aj následnosť operácií daného typu procesu, sa nijako nezaznamenávajú a tak je problém určiť, podľa akej verzie bol vykonaný konkrétny proces. V prípade doplnenia nového poľa do formuláru sa v existujúcich starších dokumentoch po otvorení detailu zobrazí nové pole s prázdnu hodnotou.

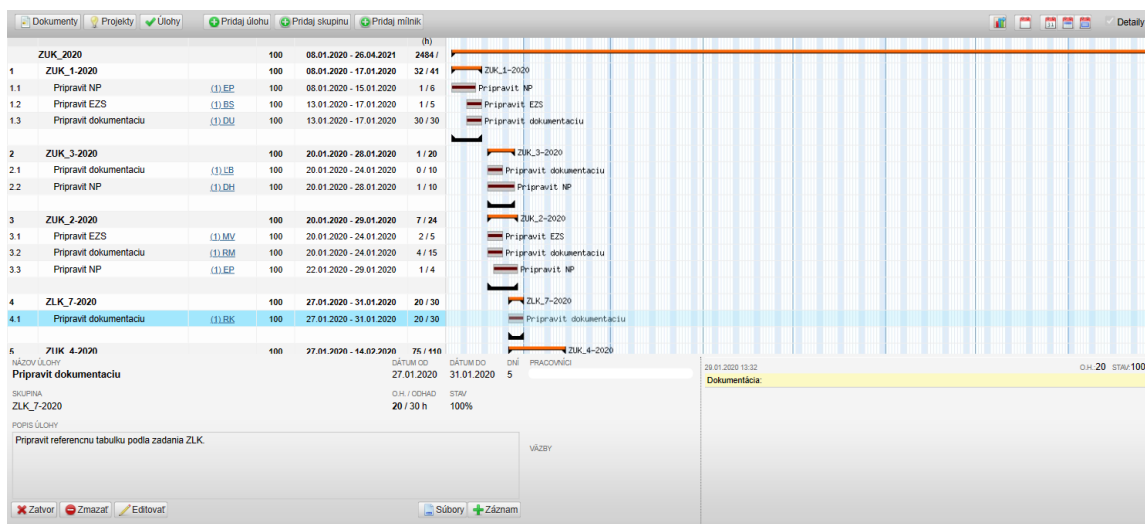
Osoba s oprávnením vytvorí dokument kliknutím na tlačidlo *Nový dokument* na domovskej stránke, vyberie typ dokumentu a vyplní úvodný formulár pre zvolený typ. Vyplnený formulár je časť „Dokument“ na obrázku 4.3. Podaný dokument najskôr zhodnotí prvotný schvaľovateľ (napríklad technický riaditeľ), ktorý môže návrh vrátiť na prepracovanie, rovno zamietnuť alebo určiť osoby na vyjadrenie, schváliť, a tým návrh posunúť ďalej. Tieto možnosti sú zobrazené na obrázku 4.8. K dokumentu potom určené osoby dodajú svoje stanoviská a ich reakcie možno vidieť v časti „Vyjadrenia“ na obrázku 4.3. Vyjadrovania môžu prebiehať vo viacerých kolách, ak je potrebné získať nové stanoviská, napríklad po doložení

Modul „Projekty“ slúži na riadenie projektov v rámci firmy. Obdobne ako modul „Dokumenty“, je aj tento modul založený na tabuľkovom zobrazení obsahu. Obrázok 4.9 ukazuje tabuľku s jednotlivými projektmi. Význam stĺpcov, ako aj princíp zafarbení, je obdobný k tým v module „Dokumenty“. Odlišné sú stĺpce *Začiatok* a *Koniec*, ktoré označujú dátum začatia, respektíve predpokladaný dátum ukončenia projektu. Stĺpec *Priebeh* zobrazuje percentuálne dokončenie všetkých úloh, a teda zobrazuje postup v projekte v rozsahu od 0 % do 100 %. Tento modul disponuje možnosťami ako napríklad rozkladanie projektu na úlohy a tie na podúlohy, sledovanie časových závislostí či plánovanie aktivít vopred na presný termín. Ukážka detailu projektu je na obrázku 4.10. Modul sa veľmi nevyužíva, nakoľko sa v priebehu 10 rokov v tomto systéme vytvorilo iba 12 projektov. Niektoré oddelenia používajú špecializované softvéry, ktoré obsahujú aj riadenie projektov, a tak uprednostňujú tieto riešenia. Medzi využívané softvéry vo firme patria napríklad Windchill, Creo⁹ alebo TPV2000 Plus¹⁰.



Označenie	Názov	Typ	Vytvoril	Začiatok	Koniec	Priebeh	Stav	Akcie
PV_4-2013	K O M P R E S O R - KYVNY PIEST	projekt vývojový		14.11.2013	13.11.2014		V plánovaní	
PV_3-2017	VÚ-05-17 ADSORBCNY SUSIC M5	projekt vývojový		18.9.2017	28.8.2018		Ukončený	
PV_1-2020	ZUK_2020	projekt vývojový		8.1.2020	26.4.2021		Spustený	
PV_5-2020	PV_2-2020	projekt vývojový		1.9.2020	30.4.2021		V plánovaní	
PV_1-2021	ZUK_2021	projekt vývojový		11.1.2021	22.12.2021		Spustený	
PV_1-2021	VU-02-19	projekt vývojový		16.4.2021	11.8.2021		V plánovaní	
POC_18-2021	CEPHEUS - ARLX900	Platená oprava CNC		15.11.2021	30.8.2022		Spustený	
PV_4-2021	CryoFIB projekt	projekt vývojový		15.11.2021	8.7.2022		Spustený	
PV_5-2021	ELMARK+ E-nabjacka	projekt vývojový		6.12.2021	14.4.2022		V plánovaní	
PV_1-2022	ZUK_2022	projekt vývojový		17.1.2022	28.2.2023		Spustený	
ZUK_12-2022	TESTOVACI PROJEKT			17.2.2022	28.2.2022		Spustený	
PV_7-2023	ZUK_2023	projekt vývojový		17.1.2023	6.12.2023		Spustený	

Obr. 4.9: Modul projektov systému Workflow.



NÁZOV ÚLOHY	DÁTUM OD	DÁTUM DO	DNE	PRACOVNÍCI	STAV
ZUK_1-2020	08.01.2020	26.04.2021	2484 /		
1 ZUK_1-2020	08.01.2020	17.01.2020	32 / 41		
1.1 Príprava NP	08.01.2020	15.01.2020	1 / 6		
1.2 Príprava EZS	13.01.2020	17.01.2020	1 / 5		
1.3 Príprava dokumentáciu	13.01.2020	17.01.2020	30 / 30		
2 ZUK_3-2020	20.01.2020	28.01.2020	1 / 20		
2.1 Príprava dokumentáciu	20.01.2020	24.01.2020	0 / 10		
2.2 Príprava NP	20.01.2020	28.01.2020	1 / 10		
3 ZUK_2-2020	20.01.2020	29.01.2020	7 / 24		
3.1 Príprava EZS	20.01.2020	24.01.2020	2 / 5		
3.2 Príprava dokumentáciu	20.01.2020	24.01.2020	4 / 15		
3.3 Príprava NP	22.01.2020	29.01.2020	1 / 4		
4 ZLK_7-2020	27.01.2020	31.01.2020	20 / 30		
4.1 Príprava dokumentáciu	27.01.2020	31.01.2020	20 / 30		
5 ZLIK_4-2020	27.01.2020	14.02.2020	75 / 110		

NÁZOV ÚLOHY	DÁTUM OD	DÁTUM DO	DNE	PRACOVNÍCI	STAV
Príprava dokumentáciu	27.01.2020	31.01.2020	5		
ZUK_7-2020	20 / 30 h				100%

Obr. 4.10: Detail projektu v systéme Workflow.

⁹Windchill a Creo sú oba softvéry od spoločnosti PTC, <https://www.ptc.com/en>

¹⁰TPV2000 Plus od spoločnosti TPV, <https://www.tpvgroup.cz/>

Modul „Úlohy“ zhromažďuje riešiteľské úlohy, ktoré vznikli v moduloch „Dokumenty“ alebo „Projekty“. Rozhranie modulu (obrázok 4.11) je konzistentné s predchádzajúcimi – vo vrchnej časti je možnosť vyhľadávania a filtrovania a zvyšok stránky je tabuľka s úlohami. Význam stĺpcov je znova analogický s významom v predchádzajúcich moduloch. Rozdielne sú stĺpce *Zadávateľ* (tvorca úlohy), *Riešiteľ* (osoba, pre ktorú je úloha určená), *WF/Projekt* (súvisiaci proces workflow alebo projekt) a *Po* (omeškanie nedokončených úloh po termíne dokončenia). Princíp sfarbenia je tiež rovnaký. Po otvorení detailu úlohy sa zobrazia informácie vo forme ako na obrázku 4.12. Vľavo možno vidieť prípadnú nadradenú úlohu, sesterské alebo podradené úlohy v stromovej štruktúre spolu s menom riešiteľa a percentuálnym dokončením úlohy. V strednej časti sa nachádzajú najmä informácie o zvolenej úlohe (vrátane zadania) a pole pre uloženie zápisu o priebehu úlohy.

Číslo	Názov	Zadávateľ	Riešiteľ	Priebeh	Začiatok	Koniec	WF/Projekt	Po ...	Akcie
4514-2014	testovacia uloha			<div style="width: 100%; background-color: green;"></div>	17.7.2014	17.7.2014		1526d	
18184-2014	test			<div style="width: 100%; background-color: green;"></div>	19.2.2021	17.7.2014		2413d	
18196-2014	test			<div style="width: 100%; background-color: green;"></div>	19.2.2021	17.7.2014		2413d	
18207-2014	test 5			<div style="width: 100%; background-color: green;"></div>	19.2.2021	17.7.2014		2413d	
18559-2014	test notifikacie			<div style="width: 100%; background-color: green;"></div>	25.3.2021	17.7.2014		2822d	
18561-2014	skuska			<div style="width: 100%; background-color: green;"></div>	25.3.2021	17.7.2014		2822d	
18562-2014	mailer			<div style="width: 100%; background-color: green;"></div>	25.3.2021	17.7.2014		2822d	
20561-2022	Z_3437-2021			<div style="width: 100%; background-color: green;"></div>	2.12.2021	8.4.2022	Z_3437-2021 ...		
PV_5-2012-...	Dokumentáciu stroj. do výroby ZUK 11-2012			<div style="width: 100%; background-color: green;"></div>	15.2.2012	27.2.2012	Zakazkove uloh...	7d	
PV_5-2012-...	Dokumentáciu stroj. do TPV ZUK 11-2012			<div style="width: 100%; background-color: green;"></div>	15.2.2012	29.2.2012	Zakazkove uloh...	5d	
PV_5-2012-...	NP do výroby ZUK 11-2012			<div style="width: 100%; background-color: green;"></div>	15.2.2012	29.2.2012	Zakazkove uloh...		
PV_5-2012-...	Dokumentáciu stroj. do výroby ZUK 10-2012			<div style="width: 100%; background-color: green;"></div>	15.2.2012	30.3.2012	Zakazkove uloh...	14d	
PV_5-2012-...	Dokumentácia stroj. do WCH ZUK 10-2012			<div style="width: 100%; background-color: green;"></div>	15.2.2012	6.4.2012	Zakazkove uloh...	10d	
PV_5-2012-...	Dokumentácia elektro do výroby ZUK 10-2012			<div style="width: 100%; background-color: green;"></div>	15.2.2012	30.3.2012	Zakazkove uloh...		
PV_5-2012-...	Dokumentácia elektro do WCH ZUK 10-2012			<div style="width: 100%; background-color: green;"></div>	15.2.2012	2.4.2012	Zakazkove uloh...		

Obr. 4.11: Zoznam úloh v module Úlohy systému Workflow.

Úlohy

100 % 02.12.2021
Z_3437-2021 - Delenie zaznamov v IS WorkFlow pre potreby PMS reportov (20560-2022) ✓

100 % 02.12.2021
Z_3437-2021 (20561-2022) ✓

100 % 02.12.2021
Z_3437-2021 (20562-2021) ✓

100 % 02.12.2021
Z_3437-2021 (20563-2021) ✓

100 % 02.12.2021
Z_3437-2021 (20564-2022) ✓

Názov úlohy: **Z_3437-2021 (20561-2022)** ✓

Názov úlohy v EN:

Riešiteľ: _____ **Zadávateľ:** _____
Dátum od: 2021-12-02 **Dátum do:** 2022-04-08
Predpokladaný počet hodín: 0 **Skutočný počet hodín:** 12
Workflow: [Z_3437-2021 - Delenie zaznamov v IS WorkFlow pre potreby PMS reportov](#)
Priebeh: 100%
Zadanie: _____
Priložené súbory: _____

[Pridať podúlohu](#)

08.04.2022 - 17:45 Priebeh: 100% Odpracovane 12 hod.

Nasadené na živom WF. Zahnuté v: G8DI,G8K,SKZ,SVK

Prílohy:

Zápis o priebehu:

Súbor 1: Nie je vybraný žiadny súbor
Súbor 2: Nie je vybraný žiadny súbor
Súbor 3: Nie je vybraný žiadny súbor

Priebeh: 100 %
Odpracované: 0 hod

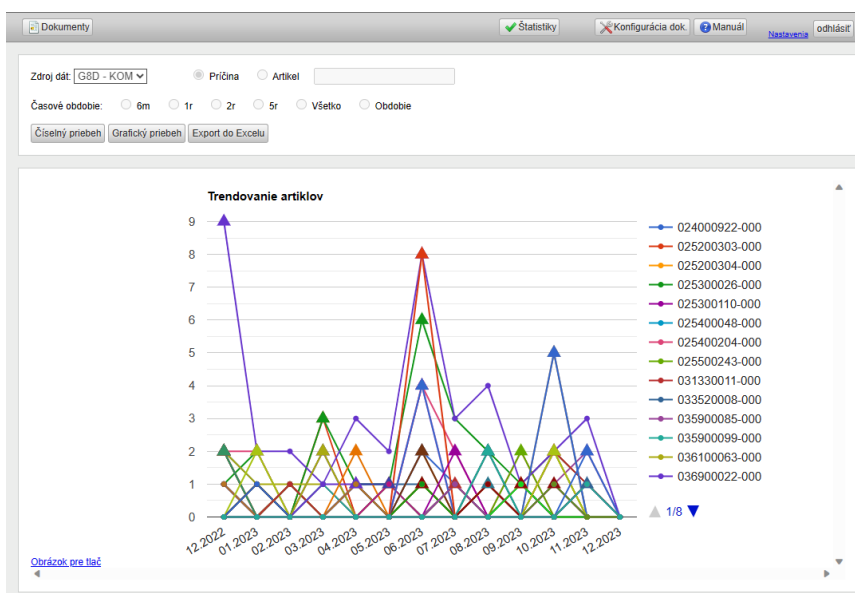
Obr. 4.12: Detail úlohy v module Úlohy.

Modul „Štatistiky“ umožňuje štatistickú prácu s dátami v systéme. Ponuka možností tohto modulu, ako aj ďalšie ukážky aktuálneho systému, sú zahrnuté v prílohe A.

Každá reklamácia zaznamenaná v dokumentoch G8D má uvedenú príčinu. Na obrázku 4.13 je prehľad počtov reklamácií zaznamenaných za posledný rok rozdelený podľa mesiacov a príčin. Obrázok 4.14 ukazuje, kolkokrát sa vyskytlo artiklové číslo v dokumentoch G8D za uplynulý rok. Na vodorovnej osi je rozdelenie do mesiacov a na zvislej osi je počet reklamácií. Legenda vpravo obsahuje artiklové čísla a k nim priradené farby čiar, ktoré v grafe značia počet reklamácií daného artiklového čísla v danom mesiaci.

Príčina	12.2022	01.2023	02.2023	03.2023	04.2023	05.2023	06.2023	07.2023	08.2023	09.2023	10.2023	11.2023	12.2023
Hlučnosť	0	1	0	0	1	0	0	0	0	0	0	0	0
Balenie - obal/preprava	0	0	0	0	0	0	0	0	0	1	0	0	0
Nekompletná dodávka	0	0	0	0	0	0	0	0	0	1	0	2	0
Porucha riadiacej jednotky	1	1	0	1	2	1	3	1	1	0	1	4	0
Porucha motora	0	1	0	0	1	1	0	0	0	1	2	0	0
Porucha el.prvky	10	4	4	3	5	4	12	4	7	6	8	5	0
Poruchy pneumatické prvky	5	3	1	6	1	1	6	8	3	4	2	3	0
Poruchy mechanické prvky	6	3	0	4	1	2	7	1	2	0	1	0	0
Kondenzácia	0	0	0	0	0	0	0	0	0	0	0	0	0
Nesprávne označenie	0	0	0	0	0	0	0	0	0	0	0	0	0
Nesprávne vyhotovenie	1	3	3	0	0	3	0	1	1	0	1	2	0
Zavinené závady zo strany zákazníka	0	0	0	0	0	0	0	0	0	0	0	0	0
Iné	0	1	0	0	0	0	0	3	0	0	0	3	0
Spolu	3	2	3	3	3	1	1	2	4	0	0	0	0

Obr. 4.13: Tabuľka s počtom príčin reklamácií v dokumentoch G8D pre divíziu Kompresory za posledný rok rozdelený po mesiacoch.



Obr. 4.14: Graf zobrazujúci počet výskytov artiklových čísel v dokumentoch G8D pre divíziu Kompresory za posledný rok.

4.1 Názory používateľov systému – dotazník

Pre porozumenie fungovania systému z používateľského hľadiska bol vytvorený dotazník určený pre zamestnancov. Obsahoval 16 otázok, ktoré sa týkali najmä názorov na používaný systém, jeho funkčnosti a spôsob používania. Dotazník bol dobrovoľný a čiastočne anonymný, teda ľudia ochotní osobne prediskutovať svoju odpoveď mohli vyplniť aj svoje meno. Dostal som 36 odpovedí z celkových približne 200 zamestnancov. Treba však spomenúť, že spoločnosť EKOM je firma, kde je väčšina pracovníkov zamestnaná vo výrobe, a teda so systémom nepracujú. V dotazníku boli otázky rôznych typov – výber jednej alebo viacerých možností a otvorené otázky. Pri otázkach s výberom som odpovede priamo previedol na hodnoty v grafe. Odpovede na otvorené otázky som si zoskupoval podľa nosných myšlienok v odpovediach a tie som previedol do grafu. Odpovede bez výpovednej hodnoty som do výsledkov nezahŕňal.

Jedna z otázok sa pýtala na pracovnú pozíciu vo firme a ukázala, že hlavnými používateľmi sú vedúci nižších organizačných jednotiek, konštruktéri a manažéri oddelení. Rozloženie pracovných pozícií respondentov je na obrázku 4.15. So systémom pracuje tiež vedenie spoločnosti a samozrejme IT správcovia.



Obr. 4.15: Otázka číslo 16: Vaša pracovná pozícia v rámci firmy

Z výsledkov dotazníku vyplýva, že väčšina respondentov využíva systém na zakladanie a vyjadrovanie sa k rôznym dokumentom a späťne si vyhľadávajú údaje v ukončených dokumentoch. Nasledujú operácie týkajúce sa úloh, konkrétne zadávanie a vykonávanie úloh. Používatelia vyzdvihujú celkovú prehľadnosť systému, sú spokojní s vyhľadávaním úloh a páčia sa im spôsoby zadávania úloh, zakladania dokumentov a vyjadrovania sa k nim. Naopak poukazujú na problematickú prácu s textovými poľami a formulármi, ktoré odstraňujú čo i len základné formátovanie textu. Výhrady majú aj voči e-mailovým notifikáciám, ktoré v tele obsahujú iba odkaz na dokument bez sprievodných informácií, napríklad označenie dokumentu. Väčšine však vyhovuje notifikácia pre každú udalosť, no nájdu sa aj takí, ktorým ich chodí príliš veľa a stačil by im sumárny výpis raz za deň.

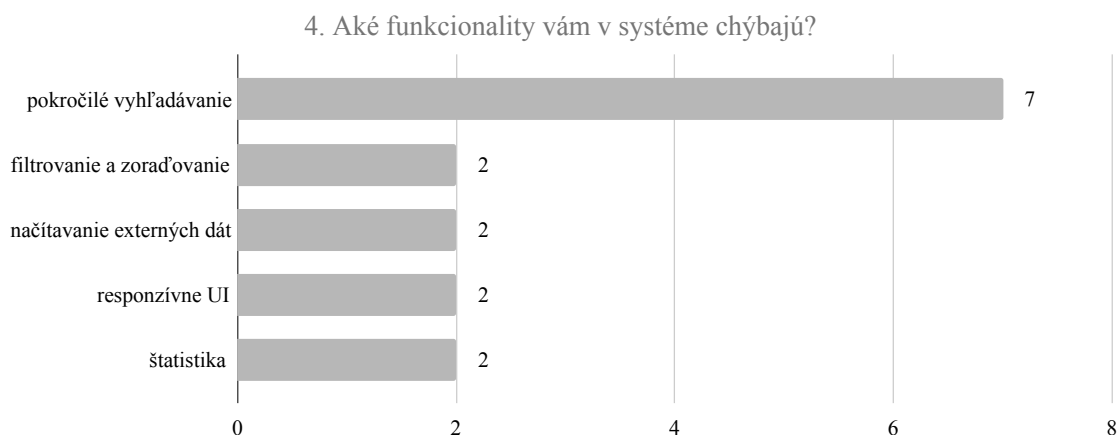
V odpovediach na obrázku 4.16 respondenti uviedli, že pozornosť by si zaslúžilo aj obnovenie používateľského webového rozhrania a zavedením princípov responzívneho dizajnu by sa systém mohol používať aj na menších obrazovkách mobilných zariadení. Ďalším želaným vylepšením by bola možnosť vytvorenia konceptu bez samotného založenia nového doku-

mentu. Osobná konzultácia s jedným z respondentov priniesla podrobnejší pohľad na túto záležitosť. Pre vytvorenie nového dokumentu sa musí vyplniť úvodný formulár pre daný typ dokumentu. Vďaka tomuto konceptu by zamestnanci mohli vyplňať formulár a v prípade potreby zozbierania dodatočných informácií počkať s jeho podaním. Fungovalo by to ako rozpísaný a neodoslaný e-mail. Okrem funkcionalít na obrázku 4.16 boli ako problematické označené limitovaný počet príloh pre úlohy či možnosti exportovania dát.



Obr. 4.16: Otázka číslo 3: Ktoré funkcionality sa vám nepáčia?

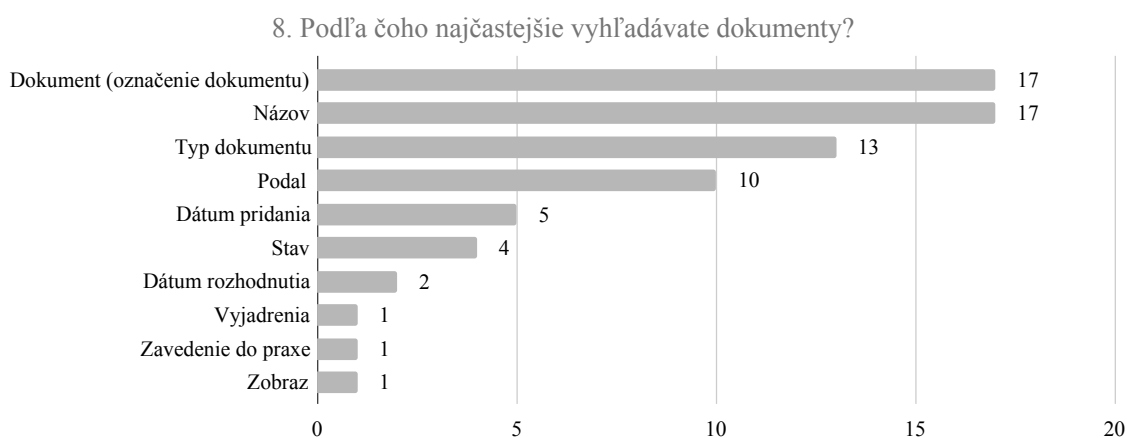
Otázka číslo 4 (obrázok 4.17) ukázala, že zamestnanci by v novom systéme uvítali pokročilé vyhľadávanie a filtrovanie podľa rôznych dát obsiahnutých vnútri dokumentov, ako napríklad artiklové číslo alebo typ výrobku. V aktuálnom systéme si musia pracne otvárať dokumenty a z obsahu zisťovať, či obsahujú potrebné informácie. Podľa dotazníku by používateľom pomohlo ponúkať dáta z iných systémov, napríklad zoznam artiklových čísel, z ktorého sa iba vyberú požadované položky. Medzi ďalšie chýbajúce funkcionality zaradili napríklad zastupovanie v neprítomnosti, pridávanie poznámok k úlohám alebo úpravy preferencií používateľského rozhrania.



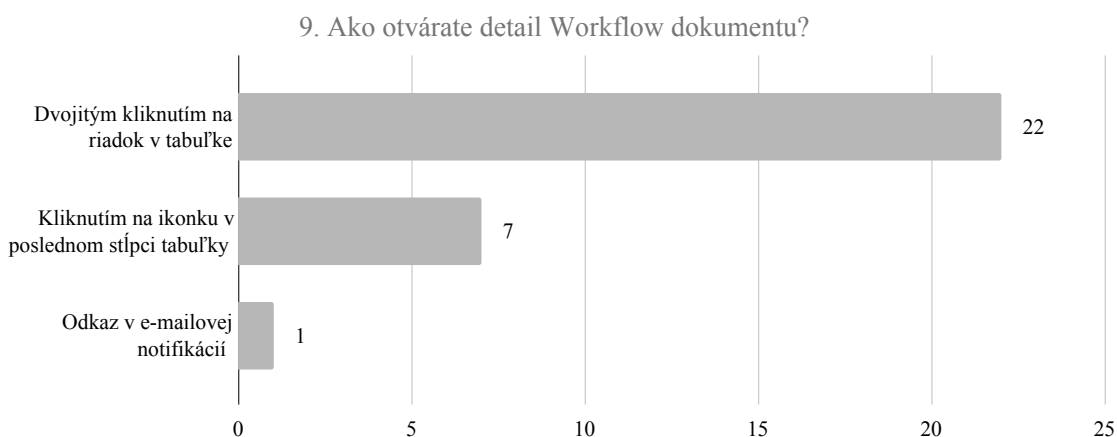
Obr. 4.17: Otázka číslo 4: Aké funkcionality vám v systéme chýbajú?

Okrem iného bola v dotazníku otázka zobrazujúca proces zapísaný v štandarde BPMN¹¹, ktorej cieľom bolo zistiť, či by sa ujalo zobrazovanie priebehu aj aktuálneho stavu vo vizuálnej podobe. Drvivá väčšina odpovedala kladne, pár odpovedí nebolo jednoznačných, no nikto neodpovedal jasne negatívne.

Snahou minimalizovať počet potrebných elementov vo webovom rozhraní som zisťoval, ktoré stĺpce tabuľky dokumentov sú pre používateľov relevantné. Zjave všetky stĺpce sú podstatné, ale niektoré sú dôležitejšie ako iné. Dominuje názov dokumentu a jeho označenie. Ako ukazuje obrázok 4.18, podľa týchto položiek zamestnanci aj najčastejšie vyhľadávajú dokumenty. Odstrániť sa môže stĺpec *Zobraz*, ktorý obsahuje ikonku na otvorenie detailu. Toto podporuje aj odpoveď na spôsob otvárania detailu dokumentu. Dvojité kliknutie na riadok v tabuľke je podľa odpovedí na obrázku 4.19 pre prevažnú väčšinu respondentov dostatočne intuitívne a samostatné tlačidlo je teda nadbytočné.



Obr. 4.18: Otázka číslo 8: Podľa čoho najčastejšie vyhľadávate dokumenty?



Obr. 4.19: Otázka číslo 9: Ako otvárate detail dokumentu workflow?

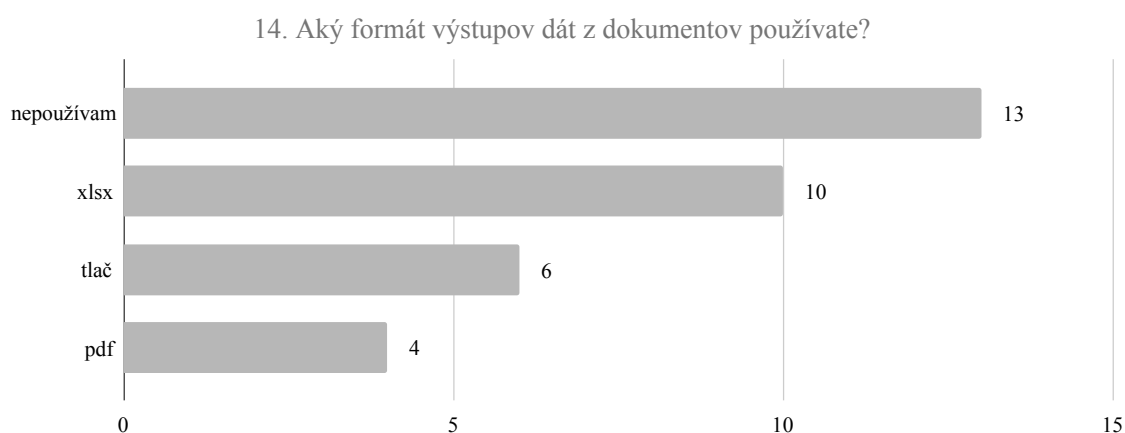
Spôsob zobrazenia zoznamu dokumentov sa javí užitočnejšie rozdeliť podľa typu do samostatných pohľadov namiesto zobrazovania jedného spoločného. Popis akcií však treba

¹¹viac na <https://www.bpmn.org/> alebo v časti 2.5

zachovať v textovej podobe s možnosťou pridružených ikoniek. Nakoľko so systémom pracuje široká škála zamestnancov, či už z pohľadu veku, pracovnej pozície alebo znalostí, tak samostatné ikonky sa nejavia ako dostatočne jasná reprezentácia akcie a bude treba zachovať aj textový popis.

Do systému sa prihlasuje prevažná väčšina a to najmä z pracovných stolných počítačov, prípadne notebookov. Len dvaja respondenti používajú systém bez prihlásenia a nikto s ním nepracuje na mobilných zariadeniach.

Štatistiky ponúkajú široké možnosti spracovania dát na rôzne využitia, no mnoho respondentov ich podľa 4.20 nepoužíva. Hlavné využitie majú na sledovanie reklamácií podľa artiklových čísel. Ak budem uvažovať iba používateľov pracujúcich so štatistikami, tak tí najviac exportujú dáta do súborov XLSX a v menšom množstve tlačia buď na papier alebo do súborov PDF.



Obr. 4.20: Otázka číslo 14: Aký formát výstupov dát z dokumentov používate?

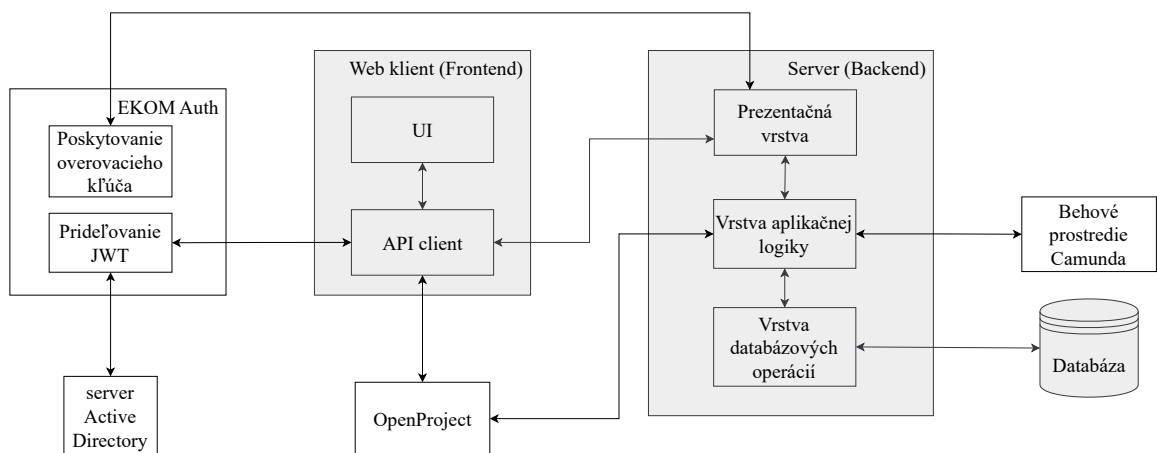
Dá sa povedať, že systém nespĺňa dnešné očakávania používateľov pre intuitívnu prácu, a to najmä obtiažnym vyhľadávaním dát v dokumentoch alebo nemožnosťou osobnej konfigurácie notifikácií či zobrazovaných údajov. Naopak sa respondentom páči jednoduché používateľské rozhranie bez zbytočných prvkov, a to aj napriek tomu, že neodráža trendy moderných webových rozhraní. Zvyšok výsledkov dotazníka je priložený v prílohe B.

Kapitola 5

Návrh nového systému

Vhodný návrh a rozdelenie systému podľa správnej architektúry je základom pre ľahšiu údržbu kódu alebo rozširovanie funkcionalít v neskorších fázach životného cyklu softvéru, najmä čo sa týka produkčnej prevádzky. Vtedy je nasadzovanie nových verzií vskutku kritickou operáciou, aby sa náhodou neznefunkčnil celý systém. Do takýchto situácií sa dostával aj aktuálne používaný systém Workflow, čomu sa pokúsím predísť už pri navrhovaní nového riešenia.

Systém potrebuje prepracovať od základov, a tak jedinými spoločnými znakmi doterajšieho a nového informačného systému môže byť jeho zameranie pre naplnenie potrieb firmy, pomenovanie a podobné používateľské rozhranie pre ľahšiu prácu s novým systémom vďaka známym prvkom. Takisto zostane zachovaný princíp práce s inštanciami procesov workflow z používateľského hľadiska, a to postupné vytváranie dokumentov. Schéma navrhovaného systému je zobrazená na obrázku 5.1. Cieľom tejto práce je implementácia podfarbených častí a komunikácia so zvyšnými jednotkami.



Obr. 5.1: Schéma navrhovaného systému Workflow pre spoločnosť EKOM.

Používateľské rozhranie nového systému zostane dostupné cez webový prehliadač, no oddelí sa prezentačná vrstva od vrstvy aplikačnej logiky. Zámerom je vytvoriť trojvrstvovú variantu architektúry klient-server popísanej v kapitole 3 a dosiahnuť vysokú úroveň rozdelenia zodpovedností medzi súčasťami systému.

Databázová vrstva bude realizovaná na dedikovaných databázových serveroch v podobe NoSQL databáze. Hlavným dôvodom je ukladanie dát v neštrukturalizovanej podobe.

Každý proces workflow zbiera iné dáta a aj podoba týchto dát sa môže v čase meniť, takže prispôsobovanie podoby tabuliek databázy pri každej zmene nie je zrovna prívetivé riešenie.

Aplikačná logika bude riešená na serveri (backend) a vnútorne bude rozdelená na tri logické vrstvy – vrstva pre prístup k dátam (vykonáva databázové operácie), vrstva aplikačnej logiky (vykonávanie operácií špecifických pre konkrétny informačný systém) a prezentačná vrstva (obsluha dotazov od klientských aplikácií). Backend bude poskytovať služby prostredníctvom API typu REST, takže komunikácia využije protokol HTTP(S). Prichádzajúce požiadavky budú prechádzať cez sériu middleware funkcií, najmä na ich filtrovanie a overovanie totožnosti a oprávnení žiadateľov.

Aby sa nemuselo vymýšľať riešenie na problém, ktoré už existuje, je stabilné a zaužívané, využije sa spolupráca so špecializovanými nástrojmi. Procesy budú definované v jazyku BPMN, ktorý je štandardizovaný a podľa dotazníka aj jednoducho pochopiteľný. Modelované procesy budú vykonávané vo voľne dostupnom softvéri Workflow Engine od spoločnosti Camunda. K tomu bude pridružený editor procesov Modeler tiež od spoločnosti Camunda. Moduly aktuálneho systému zaoberajúce sa projektmi a úlohami budú nahradené softvérom OpenProject, do ktorého sa budú systémovo zakladať nové implementačné úlohy cez poskytované API rozhranie a následný manažment týchto úloh bude prebiehať v prostredí OpenProject. Aktuálny stav plnenia úloh bude zobrazený v používateľskom rozhraní nového systému Workflow. Nástroj OpenProject sa môže využívať aj na riadenie projektov a úloh, ktoré nesúvisia s novým informačným systémom pre procesy workflow, a tak poskytnúť jednotný prostriedok pre riadenie povinností naprieč firmou.

Klientská aplikácia (frontend) bude implementovaná formou webovej aplikácie pre ľahkú dostupnosť bez potreby inštalácie dodatočných aplikácií na pracovných zariadeniach, vďaka čomu do značnej miery odpadá závislosť na konkrétnej platforme. Táto aplikácia bude sprostredkovať rozhranie medzi používateľom a samotným systémom spôsobom Client-Side Rendering, teda klientská aplikácia bude získavať serializované dáta zo servera a jej úlohou bude zobraziť tieto dáta v používateľsky prívetivej podobe.

Autorizácia používateľov bude prebiehať pomocou ich doménových kont v službe Active Directory od spoločnosti Microsoft, následkom čoho bude menší počet prihlasovacích údajov, ktoré si musia pamätať. Na tento účel sa využije centrálny autorizačný server, ktorý po úspešnom prihlásení doménovými údajmi vráti klientskej aplikácii prístupový žetón v podobe JWT¹ na následné preukazovanie identity používateľa. Tento prístupový žetón bude súčasťou každého dotazu na server, a ten si overí platnosť žetónu verejným kľúčom, ktorý získa z autorizačného servera.

Pri tvorbe nového používateľského rozhrania bude snahou minimalizovať množstvo zobrazovaného obsahu a ponechať len ten nevyhnutný pre danú obrazovku, čím by sa mala dosiahnuť vyššia prehľadnosť a zrýchliť práca so systémom. Návrh rozloženia zoznamu dokumentov je na obrázku 5.2, ktorý by mal nahradiť pohľad na zoznam dokumentov z obrázku 4.2. V záhlaví sa nachádza identifikácia systému, ponuka modulov, informácie o aktuálne prihlásenom používateľovi a možnosť zmeny jazyka. Ako vyplýva z dotazníku, rozdelenie na samostatné pohľady podľa typu dokumentu je pre používateľov veľmi žiadané. Navrhovaný obsah stránky preto začína označením aktuálne prezeraného typu dokumentov, ktorý bude zároveň slúžiť ako prepínač na iné typy dokumentov. Veľmi žiadanou funkcionalitou sú aj viaceré možnosti filtrovania, ktoré budú dostupné nižšie. Taktiež tu bude odkaz na vytvorenie nového dokumentu. Zvyšok stránky vyplní tabuľka dokumentov, ktorá bude





¹JWT: skratka pre JSON Web Token, viac na <https://jwt.io/>

v porovnaní s aktuálnou podobou (ako je na obrázku 4.2) podfarbená jednotnou farbou a stav spracovania môže byť indikovaný v krúžku na začiatku riadku.

(Stav)	Označenie	Názov	Podal	Vyjadrenia	Dátum pridania	Dátum rozhodnutia	Dátum zavedenia do praxe
<input type="radio"/>	Z_0000-2024	Test	Jožko Tester	0/0	1.1.2024		
<input type="radio"/>	Z_0000-2024	Test	Jožko Tester	0/0	1.1.2024		
<input type="radio"/>	Z_0000-2024	Test	Jožko Tester	0/0	1.1.2024	1.1.2024	
<input type="radio"/>	Z_0000-2024	Test	Jožko Tester	0/0	1.1.2024	1.1.2024	
<input type="radio"/>	Z_0000-2024	Test	Jožko Tester	0/0	1.1.2024	1.1.2024	1.1.2024

Obr. 5.2: Zoznam dokumentov typu *Návrhy na odchýlku* v novom systéme.

Aktuálny systém má ako domovskú stránku priamo zoznam dokumentov a typy procesov zobrazuje v malom výberovom poli. Pre sprehľadnenie zoznamu podnikových procesov vo firme a zníženie množstva prenášaných dát už pri prvom načítaní stránky som navrhol domovskú stránku v podobe rázcestníka na obrázku 5.3. Ten ukazuje kategorizované typy dokumentov, kde každý odkazuje na zoznam dokumentov zvoleného typu (ako na obrázku 5.2). Typy obsiahnuté v obrázku vychádzajú z aktuálne používaných typov a sú roztriedené podľa divízie uplatnenia (CNC a KOM), G8D má vlastnú kategóriu, nakoľko sa jedná o významovo ucelenú jednotku a medzi ostatné sú zaradené tie typy, ktoré sa nedajú jednoducho priradiť k miestu uplatnenia.

 CNC	 KOM	 G8D	 Ostatné
Zmeny Odchýlky Zákazková úloha Reklamácia dodav Platená oprava G8D report	Zmeny Odchýlky Zákazkový list Spätná väzba G8D report	KOM G8D report CNC G8D report DI G8D report	Vestník noriem Nový nakupovaný artikel Všetky

Obr. 5.3: Návrh domovskej stránky s rázcestníkom a ponukou rôznych typov dokumentov.

Kapitola 6

Implementácia navrhnutého systému

Pri výbere vhodných nástrojov na vytvorenie nového systému vychádzam z poznatkov získaných analýzou súčasného riešenia, požiadaviek firmy, ako aj z názorov používateľov, ktoré som obdržal v rámci dotazníku a zároveň sa snažím pridržať navrhnujej schémy systému z obrázku 5.1.

Primárnym programovacím jazykom je TypeScript, čo predstavuje nadmnožinu jazyka JavaScript, ktorý dopĺňa najmä o statické typovanie. Amr Saafan [27] vidí výhodu jazyka TypeScript najmä v silnom typovaní premenných a parametrov funkcií vynucovanom pri vývoji a následnej kompilácii kódu. To pomáha s odhaľovaním chýb po celú dobu vývoja programu, dôsledkom čoho je menej chýb pri produkčnom používaní programu, kedy je oprava chýb o niečo náročnejšia. Kód napísaný v jazyku TypeScript sa vďaka silnému typovaniu tiež čiastočne sám dokumentuje, čo urýchľuje pochopeniu kódu ďalšími osobami alebo aj samotným autorom, ktorý sa po dlhšom čase vráti k tomuto kódu. Moderné vývojové prostredia navyše dokážu poskytovať rady pre vývojárov podľa používaných dátových typov. Okrem iného jazyk TypeScript možno využiť pri programovaní webových aplikácií, ako aj pri programoch pre behové prostredie Node.js, z čoho plynie výhoda udržiavania rovnakej konzistencie kódu naprieč súčasťami aplikácie. Základnú funkcionálnu jazyka dopĺňam rôznymi voľne dostupnými balíčkami pochádzajúcimi z registra *npm*¹.

6.1 Databázová vrstva

Na uchovávanie dát som zvolil nerelačnú databázu MongoDB², ktorá ukladá dáta v podobe takzvaných „dokumentov“, kde štruktúra obsahu nie je pevne daná. Značná podobnosť s objektami formátu JSON prispieva k jednoduchej práci s dátami.

S databázou MongoDB nepracujem priamo, ale využívam objektovo-dokumentové mapovanie (*ODM*) balíčka *Mongoose* pre aplikácie bežiacie v prostredí Node.js. Prostredníctvom tohto nástroja taktiež definujem základnú formu dát uchovávaných v databázi v podobe schém modelov v priečinku `server/src/DAL/schemas/`. Databázové operácie nad databázou MongoDB sú definované v priečinku `server/src/DAL/` v súboroch končiacich na `...Repository.ts`, čím sa tieto operácie oddeľujú od zvyšku kódu a dodržiavajú navrhnuté architektonické rozdelenie – teda dátovú vrstvu.

¹Npm je nástroj pre správu balíčkov v jazyku JavaScript, viac na <https://www.npmjs.com/>

²Domovská stránka databázy MongoDB: <https://www.mongodb.com/>

V tejto databáze ukladám dáta o počítačových dokumentoch, inštanciách procesov, používateľských formulároch, ako aj o definíciách procesov. Behové prostredie Camunda si ukladá definície procesov a formulárov aj interne, no ponúka sledovanie verzií iba definícií procesov. Inšpiroval som sa poskytovanou špecifikáciou OpenAPI k tomuto systému a konzistentne s ňou som použil označenie *latestVersion* pre označenie najaktuálnejšej verzie procesu, respektíve používateľského formuláru. Aktualizácia tejto položky je vykonaná po úspešnom nahrať nových definícií do prostredia systému Camunda, kedy sa zmení tento príznak na nepravdivú hodnotu a v databáze sa vytvorí záznam pre novú verziu.

Nakoľko mi prišlo používateľsky prívetivejšie vyhľadávanie dokumentov, teda inštancií procesov, podľa ich firemného označenia (napríklad *Z_1234-2024*) namiesto náhodného čísla alebo reťazca znakov, je potrebné k efektívnemu vyhľadávaniu prispôbiť samotnú databázu. Na položke *ekomId* v schéme pre inštancie procesov je definovaný index, vďaka ktorému sa zníži doba vyhľadávania jedného záznamu pri veľkom množstve záznamov. Jeden dokument v databáze MongoDB pre inštanciu procesu obsahuje identifikátor priradený databázou (*_id*), firemné označenie (*ekomId*) a pre sledovanie spätných väzieb na platformu Camunda aj identifikátor *camundaId*. Firmené označenie je pri zakladaní novej inštancie procesu generované na serveri Workflow podľa počítadla a predpony označenia priradeným k definícií procesu. Kompletná podoba databázovej schémy definovanej pre balíček *Mongoose* je vo výpise 6.1. Každý záznam obsahuje dátumy vytvorenia, poslednej úpravy a ukončené procesy majú aj dátum dokončenia. Sú tu aj položky pre vlastné metadáta (*nadpis*, *stav*, *autor*) a ďalšie metadáta zo systému Camunda (*processDefinitionId*, *processDefinitionKey*).

```
{
  _id: Types.ObjectId;
  camundaId: string;
  processDefinitionId: string;
  processDefinitionKey: string;
  createdBy: string;
  title: string;
  state: string;
  ekomId: string;
  processTasks: Types.DocumentArray<IProcessUserTask>;
  openProjectRootTaskId?: number;
  createdAt: Date;
  updatedAt: Date;
  finishedAt?: Date;
}
```

Výpis 6.1: Podoba inštancie procesu pre balíček Mongoose.

Životnosť dát ukončených inštancií procesov je v systéme Camunda časovo obmedzená, aby si interná databáza udržiavala rozumnú výkonnosť. Pre firmu dôležité dáta o každej inštancií, konkrétne formuláre vyplnené používateľmi, na základe ktorých sa vykonávajú rozhodnutia, ukladám spolu s touto inštanciou. Jednotlivé formuláre sú ukladané v položke inštancie *processTasks*, kde sa v poli dokumentov databázy MongoDB ukladajú záznamy v rovnakom poradí, ako boli vykonané.

Požiadavkou bolo aj sledovanie verzií definícií procesov. Keďže každá inštancia procesu si uchováva informáciu o tom, podľa ktorej verzie prebieha (položka *processDefinitionId*), je vhodné mať rýchly prístup k tejto definícií. Aby sa opäť zachovala konzistencia so sys-

témom Camunda, záznamy pre definície procesov zálohovaných v databáze MongoDB obsahujú položku *camundaId*, čo je identifikátor definície procesu vygenerovaný systémom Camunda. Na tejto položke je tiež nastavený index, aby bolo možné rýchlo prehľadávať množstvo definícií procesov, napríklad pri potrebe zobraziť diagram BPMN, podľa ktorého bola daná inštancia vykonaná. Položky definície procesu pre balíček Mongoose sú vo výpise 6.1. Záznam definície procesu tiež obsahuje dodatočné metadáta zo systému Camunda (*camundaKey*, *version*) a položky pre použitie vo firme s predponou *ekom*. Diagram procesu sa ukladá v položke *bpmnDiagram* ako reťazec, ktorý obsahuje jeho podobu v zápise XML.

```
{
  _id: Types.ObjectId;
  camundaId: string;
  camundaKey: string;
  name: string;
  version: string;
  latestVersion: boolean;
  description?: string;
  bpmnDiagram: string;
  ekomIdPrefix: string;
  ekomIdCounter: Types.ObjectId;
  ekomDivision: "KOM" | "CNC" | undefined | null;
  readonly createdAt: Date;
  readonly updatedAt: Date;
}
```

Výpis 6.2: Podoba definície procesu pre balíček Mongoose.

6.2 Aplikačná vrstva – server

Backend nového systému Workflow pre spoločnosť EKOM je postavený na aplikačnom rámci *Fastify* a je určený pre behové prostredie Node.js. Fastify je základom pre tvorbu aplikačného rozhrania poskytovaného serverovou časťou ďalším službám, v tomto prípade webovému klientovi informačného systému. Rozhranie štýlu REST vytvorené aplikačným rámcom Fastify predstavuje prezentačnú logickú vrstvu aplikačnej logiky, špecializujúcu sa na spracovávanie dotazov HTTP. Využívam pritom princípov zásuvných modulov a zapúzdrovania, čo sú jedny zo základných princípov aplikačného rámcu Fastify [11]. Aplikačný rámec Fastify je možné rozšíriť množstvom oficiálnych či komunitných zásuvných modulov.

Na preukázanie totožnosti používateľa využívam žetóny JWT, ktoré okrem štandardizovaných položiek z [16] obsahujú základné informácie o konkrétnom používateľovi ako prihlasovacie meno alebo zoznam skupín, do ktorých používateľ patrí. Jeden zo zásuvných modulov pridáva operácie na prácu s týmito žetónmi. Tento modul je použitý na overovanie platnosti žetónu JWT každej požiadavky pomocou verejného kľúča, ktorý môže byť získaný aj asynchrónnym volaním funkcie, a teda aj zo servera EKOM Auth. Uvažujem aj s možnosťou obmeny kľúčov na autorizačnom serveri, a to tak, že pri prvom neúspešnom overení žetónu JWT si zo servera EKOM Auth vyžiadam najnovší verejný kľúč a skúsím to znova. Ak je žetón stále neplatný, server odmieta obsluhu žiadosti. Takto je zabezpečená automatická aktualizácia verejného kľúča.

Autorizácia operácií, teda kontrola oprávnení, je riešená priamo na koncových bodoch. V aktuálnom stave odlišujem iba používateľov, ktorí sú zaradení do skupiny *admin*, teda

položka *groups* v ich žetóne JWT obsahuje aj túto hodnotu. Používatelia s touto skupinou môžu vykonávať administráciu systému – manipulovať s definíciami procesov, používateľskými formulármi či počítadlami dokumentov. Ostatným používateľom je odoslaná odpoveď so statusom HTTP 403 – nemajú oprávnenie na takú operáciu.

V súčasnom systéme sú všetky vykonávané procesy pevne definované v rámci kódu programu, čo sťažuje definíciu nových procesov či ich modifikáciu. Implementácia vlastného behového prostredia pre procesy workflow zapísaných v štandarde BPMN zahŕňa množstvo vzorov, ktoré sa v týchto procesoch vyskytujú, pričom viaceré nie sú vôbec triviálne. Čo všetko by to zahŕňalo, je uvedené v [4]. Navyše, vlastná implementácia je rozsahom na samostatnú bakalársku prácu, ako to bolo v prípade [35].

Ako bolo navrhnuté v kapitole 5, použije sa komunitná verzia softvéru Camunda Platform 7, ktorá zastrešuje všetko potrebné pre vykonávanie procesov. Najnovšou verziou je Camunda 8, ktorá je však architektonicky komplexnejšia a práca so systémom mimo programov napísaných v jazyku Java je zložitejšia. Camunda Platform 7 naproti tomu poskytuje jedno presne zdokumentované rozhranie API štýlu REST v podobe OpenAPI špecifikácie. Prostredníctvom takto definovaného rozhrania komunikuje s behovým prostredím Camunda výhradne hlavný server. Využitím balíčka *openapi-typescript* generujem z OpenAPI špecifikácie kód v jazyku TypeScript na typovo bezpečnú komunikáciu. Nakoľko nevyužívam kompletnú ponúkanú funkcionálnosť softvéru Camunda, s vybranými koncovými bodmi komunikujem prostredníctvom metód triedy *WorkflowEngineAdapter*.

Podobný prístup je zvolený aj v prípade systému OpenProject, kde server systému Workflow tiež vystupuje v pozícii klienta. Volanie koncových bodov tohto systému je abstrahované do metód triedy *OpenProjectAdapter*. S oboma systémami prebieha komunikácia prostredníctvom balíčka *openapi-fetch*. K nemu mám definované vlastné rutiny typu middleware, ktoré využívam napríklad na reagovanie na chybové odpovede. Metódy oboch spomenutých tried sú volané z kódu aplikačnej logiky, ktorá sa nachádza v priečinku *server/src/BL/*.

Za použitia balíčku *camunda-external-task-client-js* sa viem dopytovať na udalosti v behovom prostredí Camunda a reagovať na ne vlastnými funkciami napísanými v jazyku TypeScript. V novom systéme je sledovanie týchto udalostí a reagovanie na ne implementované v priečinku *server/src/camundaExternalTaskClient/*.

Nateraz server Workflow reaguje na 2 udalosti behového prostredia. Úlohou udalosti *openProjectTask* je zakladať nové implementačné úlohy v softvéri OpenProject, pričom v premenných inštancie procesu očakáva položky *rootTaskName* pre názov koreňovej úlohy, *rootTaskAssignee* ako prihlasovacie meno hlavného riešiteľa a *dueDate* pre stanovenie termínu dokončenia. Udalosť *processFinished* slúži na zmenu stavu inštancie procesu z ACTIVE na FINISHED, čo sa vo webovej aplikácii prejaví zmenou zafarbenia indikátora stavu v zozname dokumentov. Udalosti sa konfigurujú v definícii procesu na elemente aktivity *Service Task*, respektíve na elemente koncovej udalosti procesu vo variante *správa* (anglicky *Message End Event*), kde treba nastaviť typ implementácie na *External* a položke *Topic* priradiť názov jednej z udalostí behového prostredia. Obslužné funkcie pre ďalšie udalosti bude možné doimplementovať podľa potreby.

Jednou z chýbajúcich funkcionálností podľa otázky 4 vykonaného dotazníka (obrázok 4.17) bolo načítavanie externých dát v poliach formulárov. Túto logiku som implementoval v triede nazvanej *externalDataRetriever*, kde rekurzívne prechádzam komponenty žiadaného formuláru a naplňam ich externými dátami, a takto pripravený formulár sa vracia používateľovi. Na identifikáciu polí, ktoré potrebujú takéto spracovanie, pri modelovaní formuláru využívam položku *Custom Properties* na definovanie rozširujúcich vlastností pre daný element.

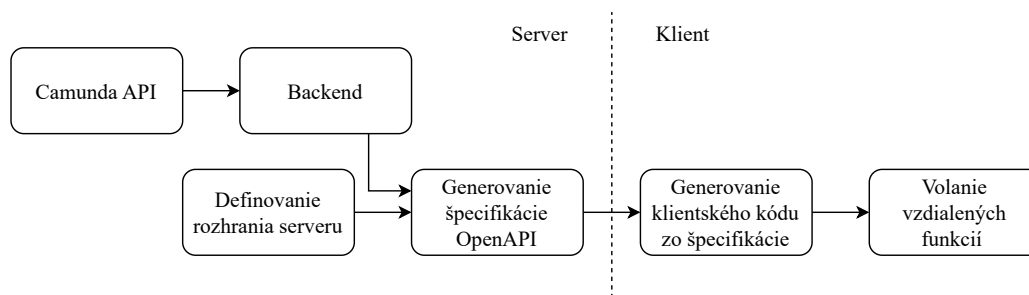
Tieto vlastnosti majú formát „klúč-hodnota“, ktoré v mojom prípade nastavujem na klúč *externalData* a hodnotu na identifikátor, podľa ktorého trieda `externalDataRetriever` získava dáta. Moja implementácia poskytuje krátky zoznam artiklových čísel pre identifikátor *articleNumbers* a zoznam používateľov, získaný zo serveru EKOM Auth, pre identifikátor *users*. Spracovávanie ďalších identifikátorov bude potrebné naprogramovať.

Aplikačný rámec Fastify je rozšírený aj o modul pre spracovanie žiadostí s typom obsahu vo formáte *multipart*, ktorý na koncovom bode `/fileServer` spracováva nahrávané prílohy k formulárom vyskytujúcim sa v procesoch. Prílohy ukladám pod priečinkom `server/uploads/`. Pri určovaní cieľovej cesty priečinku pre inštanciu procesu vychádzam z označenia vo formáte `<prefix>_<poradové číslo>-<rok>`. Toto označenie rozdeľujem na jednotlivé položky, z ktorých späťne skladám cieľovú cestu priečinku konkrétnej inštalácie v podobe `uploads/<prefix>/<rok>/<poradové číslo>`. Zámerom je zamedziť tvorbu priečinkov všetkých inštancií v koreňovom priečinku, ale čo možno v najväčšej miere znížiť množstvo priečinkov na jednej adresárovej úrovni.

Ak server pri nahrávaní súboru zistí, že súbor s rovnakým názvom pre danú inštanciu procesu už existuje, novo nahrávanému súboru pridá koncovku v podobe reťazca UUID vygenerovaného knižnicou *Crypto*, ktorá je štandardnou súčasťou behového prostredia Node.js. Univerzálne unikátny identifikátor, anglicky *Universally Unique Identifier*, je reťazec s dĺžkou 128 bitov, ktorý sa používa na garantovanie unikátnosti identifikátora [19]. Takto môžu existovať viaceré súbory s podobným názvom a používateľ nie je obťažovaný s potrebou premenovania nahrávaných súborov. Pre zaznamenávanie väzby medzi formulárom a nahrávanými súborami sa v databázovej položke inštalácie procesu `processTasks` doplní pole `appendices` názvami nahratých súborov.

Server poskytuje rozhranie štýlu REST, kde sú koncové body popísané schémou, podľa ktorej sa automaticky validujú obdržané dotazy. V prípade zlyhania validácie server odpovedá správou so statusom HTTP 400 – *Bad Request*, teda žiadosť, respektíve jej obsah, nespĺňa požadované vlastnosti.

Pre obmedzenie množstva chýb pri vývoji klientskej časti využívam generovanie špecifikácie rozhrania vo formáte OpenAPI, ktorá je zostavená zo schém koncových bodov. Z tejto špecifikácie následne na strane webového klienta generujem klientský kód pre volanie jednotlivých koncových bodov využitím balíčku *openapi-typescript*, ktorý je typovo bezpečný a poskytuje rady vo vývojovom prostredí. Vizualnú reprezentáciu postupu od definovania schém koncových bodov na serveri až po ich volanie zo strany klienta v systéme Workflow ponúka obrázok 6.1. Vďaka udržiavaniu špecifikácie aplikačného programového rozhrania serveru Workflow vo formáte OpenAPI bude možné, v prípade potreby vyššieho výkonu, nahradiť implementáciu serverovej časti v inom jazyku. Definície koncových bodov serveru sa nachádzajú v priečinku `server/src/API/`.

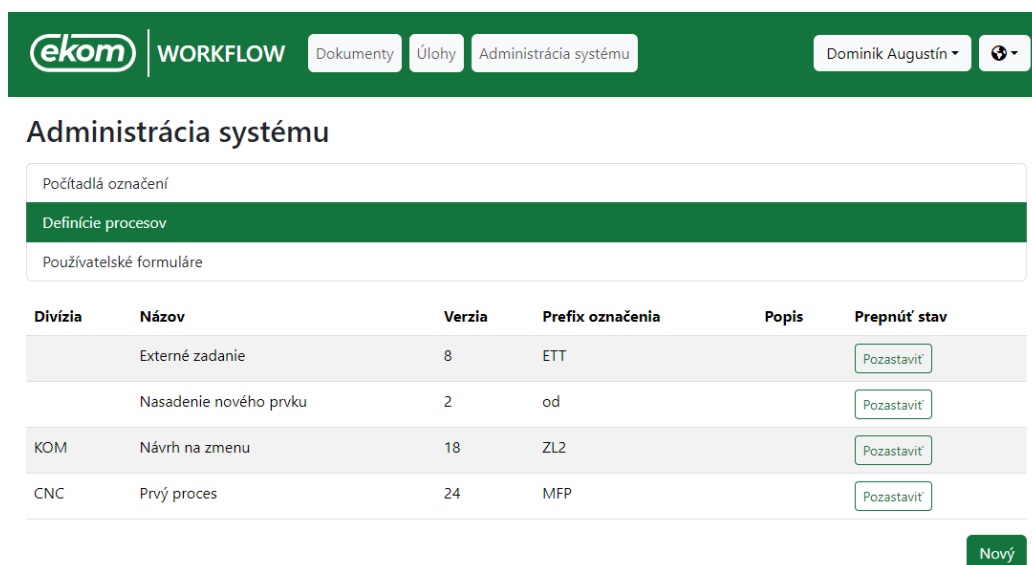


Obr. 6.1: Vizualná podoba tvorby a využitia špecifikácie OpenAPI.

6.3 Používateľské rozhranie

Klientská časť je implementovaná v podobe webovej aplikácie postavenej na aplikačnom rámci Vue.js plne kompatibilnom s programovacím jazykom TypeScript. Vue.js je založené na tvorbe vlastných komponentov, ktoré majú vlastné správanie, môžu sa vzájomne zanárať a opakovane používať. Komponenty vytváram za použitia *Composition API*, čo je odporúčaný spôsob pri použití jazyku TypeScript, v podobe takzvaných *single-file components*, ktoré obsahujú HTML šablónu, logiku a štýly CSS pre jeden komponent v jednom súbore s príponou *.vue* [34]. Tento spôsob sprehľadňuje organizáciu kódu, nakoľko všetky spolu súvisiace časti týkajúce sa jedného komponentu sú v jednom súbore, kde je kód stále ľahko uchopiteľný. Vytvorené komponenty používam v rámci ďalších komponentov alebo z nich priamo vyskladávam jednotlivé pohľady. Komponenty sa nachádzajú v priečinku `client/src/components/` a jednotlivé pohľady zase v priečinku `client/src/views/`.

Na modernizáciu používateľského rozhrania sa využíva open-source knižnica Bootstrap, vďaka ktorej sa dajú jednoducho vytvárať responzívne rozhrania za použitia webových technológií. So systémom sa bude dať ľahšie pracovať aj na mobilných zariadeniach, hoci hlavnou platformou zostávajú stolné počítače. Nakoľko nový informačný systém zostane internou aplikáciou firmy EKOM, prvoradá je funkčnosť systému a unikátny vzhľad nie je dôležitý. Podobnosť s inými stránkami využívajúcimi knižnicu Bootstrap teda nebude vadiť, naopak, jednotný štýl používateľských rozhraní môže prispieť k jednoduchšej práci so systémami. Na väčšinu zobrazovaných elementov používam štýly inšpirované ukážkami v dokumentácii knižnice [7], no niektoré časti používateľského rozhrania sú izolované balíčky, čo podliehajú vlastnému štylovaniu, a na niektorých miestach to môže byť viditeľné. Obrázok 6.2 ukazuje štylizáciu knižnicou Bootstrap v administrácii systému. V záhlaví je logo firmy s pomenovaním systému, navigácia, možnosti pre prihláseného používateľa a prepínač lokalizácie. Nižšie sa nachádza pomenovanie aktuálneho umiestnenia v systéme, tlačidlá na výber prvkov systému na konfiguráciu a samotný tabuľkový zoznam položík vybraného prvku – tu *Definície procesov*. Tabuľka obsahuje základné informácie o položkách a, v tomto prípade, možnosť pozastaviť vytváranie nových dokumentov daného procesu. V spodnej časti je tlačidlo pre pridanie novej položky.



The screenshot shows a web application interface for 'Administrácia systému'. At the top, there is a dark green navigation bar with the 'ekom' logo, the text 'WORKFLOW', and three menu items: 'Dokumenty', 'Úlohy', and 'Administrácia systému'. On the right side of the navigation bar, there is a user profile 'Domínik Augustín' and a globe icon for localization. Below the navigation bar, the main content area has a title 'Administrácia systému' and a sidebar menu with items: 'Počítadlá označení', 'Definície procesov' (highlighted in green), and 'Používateľské formuláre'. The main content area displays a table with the following data:

Divízia	Názov	Verzia	Prefix označenia	Popis	Prepnúť stav
	Externé zadanie	8	ETT		<button>Pozastaviť</button>
	Nasadenie nového prvku	2	od		<button>Pozastaviť</button>
KOM	Návrh na zmenu	18	ZL2		<button>Pozastaviť</button>
CNC	Prvý proces	24	MFP		<button>Pozastaviť</button>

At the bottom right of the table area, there is a green button labeled 'Nový'.

Obr. 6.2: Štylizácia rozhrania zobrazujúca administratívnu časť.

S aplikačným rámcom Vue.js úzko spolupracuje balíček *vue-router*. Balíček *vue-router* využívam na smerovanie v klientskej aplikácii, kde definujem adresy URL a k nim prislúchajúce pohľady, vytvorené z komponentov Vue.js, ktoré sa majú zobraziť. Tento balíček umožňuje vykonávanie funkcií v rôznych stavoch smerovania, čo využívam tak, že pred načítaním stránky kontrolujem prítomnosť žetónu JWT, jeho expiračnú dobu a v prípade administratívnych stránok aj to, či má používateľ prístup k týmto dátam. Konfigurácia smerovania sa nachádza v priečinku `client/src/router/`.

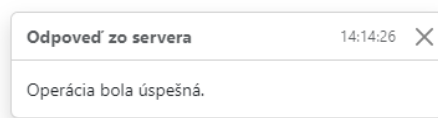
Žetón JWT ukladám v lokálnom úložisku prehliadača, s ktorým pracujem prostredníctvom balíčka *pinia*. Ten je plne kompatibilný s aplikačným rámcom Vue.js a zameriava sa na správu úložísk a aplikačných stavov v prehliadači. Pre ukladanie žetónu JWT a ďalších údajov o aktuálne prihlásenom používateľovi je definované úložisko Pinia s názvom `loggedInUser`. Pri načítaní aplikácie sa inicializuje toto aplikačné úložisko, z lokálneho úložiska prehliadača sa získa záznam uložený pod rovnomenným kľúčom a hneď sa skontroluje expirácia žetónu. V prípade exspirovaného žetónu sa záznam z lokálneho úložiska odstráni, na čo *vue-router* zareaguje presmerovaním používateľa na prihlasovaciu stránku. Používateľ sa prihlasuje svojím doménovým prihlasovacím menom a heslom cez server EKOM Auth, ktorý po úspešnom prihlásení vracia nový žetón JWT. Ten sa ukladá prostredníctvom úložiska Pinia, kde je pripravený na ďalšie využitie. Správa používaných úložísk sa sústreďuje do priečinku `client/src/stores/`.

Server EKOM Auth nie je náplňou tejto práce, a tak je na účely demonštrácie fungovania nového systému v rozsahu tejto práce nahradený napodobeninou s obdobným rozhraním. Všetky súbory týkajúce sa tejto napodobeniny, vrátane pripravených používateľských kont, sú v priečinku `authServerMock/`. Takýto prístup odľahčuje nový informačný systém o potrebu správy používateľov, ktorá je teda riadená centrálnie v službe Active Directory a v kombinácii so serverom EKOM Auth umožňuje ľahké zavádzanie ďalších systémov s autentizáciou na princípe žetónov JWT.

Na prístup k dátam na koncových bodoch serverovej časti používam spomínaný vygenerovaný kód balíčkom *openapi-typescript* podľa špecifikácie OpenAPI, ktorý aj v tomto prípade dopĺňa balíček *openapi-fetch* na využívanie tohto vygenerovaného kódu. Vďaka možnosti definovania rutiny typu middleware pre odchádzajúce požiadavky mi stačí na jednom mieste definovať pridávanie žetónu JWT pre identifikáciu aktuálne prihláseného používateľa. Podobnou rutinou middleware reagujem na obdržané odpovede a v prípade výskytu chyby vytváram notifikáciu pre používateľa. Notifikácie zobrazujem aj v prípade úspešne dokončenej operácie, ak sa nejednalo o požiadavku HTTP metódy GET, teda najmä pri manipuláciách v administratívnej časti. Triedy pre komunikáciu so serverom cez rozhranie REST sú združené v priečinku `client/src/apiClient/`.

Notifikácie v používateľskom rozhraní sa zobrazujú v pravom hornom rohu a na ich správu je vytvorené ďalšie aplikačné úložisko Pinia nazvané `notifications`. Toto úložisko má podobu fronty, do ktorej vkladám nové notifikácie a tie sa po predvolených piatich sekundách z fronty odoberú. Na samotné zobrazovanie sú dedikované dve komponenty – `NotificationToastsOverlay.vue` a `NotificationToast.vue`, kde prvá z nich vytvára kryciu vrstvu nad používateľským rozhraním s obálkou na notifikácie a druhá z nich predstavuje jednu inštanciu notifikácie podľa komponentu `Toast` z knižnice Bootstrap. Výsledkom je notifikácia ako na obrázku 6.3.

Používateľom sa podľa otázky číslo 3 (obrázok 4.16) v aktuálnom systéme nepáči, že si nevedia uložiť rozpisovaný dokument. Nový systém ponúka ukladanie každého rozpracovaného formulára, ku ktorým sa používatelia môžu neskôr vrátiť a dáta im budú automaticky načítané. Ukážka rozpracovaného úvodného formuláru je na obrázku 6.4. V hornej časti



Obr. 6.3: Ukážka systémovej notifikácie.

pod označením *Typ dokumentu* je možnosť výberu jedného z procesov definovaných v systéme. Nižšie sa zobrazuje vyplnený formulár určený pre založenie novej inštancie procesu *Návrh na zmenu* divízie *KOM*. Formulár bol vytvorený v časti *Používateľské formuláre* administrácie systému. Používateľ môže založiť nový dokument stlačením tlačidla *Odoslať* alebo si uložiť hodnoty vstupných polí na neskôr pomocou tlačidla *Uložiť koncept*.

Typ dokumentu

KOM - Návrh na zmenu

Navrh na zmenu

nazov

Demo proces

Popis súčasneho stavu

Doterajší systém Workflow.

Popis navrhovaného stavu

Nový systém Workflow.

Dotknute artikle

4202620A5-400 x 603014336-000 x 603014336-001 x Search

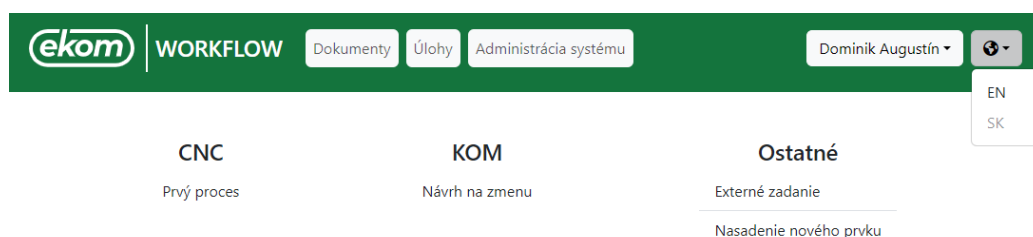
Odoslať Uložiť koncept

Obr. 6.4: Zakladanie nového dokumentu s možnosťou odoslania a uloženia ako koncept.

Požiadavku firmy na dostupnosť používateľského rozhrania minimálne v slovenčine a v angličtine som splnil využitím balíčka *i18n* vo variante pre aplikačný rámec *Vue.js*. Rozbalený prepínač medzi spomínanými jazykmi sa na obrázku 6.5 nachádza na pravom okraji. Dostupné sú možnosti *EN* pre angličtinu a *SK* pre slovenčinu. Prepnutím jazyka sa stránka nanovo načíta, aby sa aplikovala zmena. Keďže sa pri vývoji nemenili popisky stĺpcov v zozname dokumentov a v zozname procesných úloh, ktoré sú realizované balíčkom *ag-grid-vue3*, obnova stránky je nateraz potrebná. Ako predvolená lokalizácia je nastavená slovenčina, ale posledne zvolený jazyk sa ukladá do lokálneho úložiska, aby sa pri ďalšom otvorení webového klienta aplikovala táto voľba.

Lokalizované texty ukladám v súboroch JSON v priečinku `client/src/locales/`. Ich obsah je štrukturovaný v podobe vnorených objektov JSON podľa cieľového miesta použitia a samotné lokalizované texty predstavujú atribúty objektu. V šablónach komponentov aplikačného rámca *Vue.js* prístupujem ku konkrétnemu textu volaním globálne dostupnej funkcie, ktorej argumentom je prístupová cesta k položke v objekte JSON, napríklad `$t("appHeader.navigationMenu.documents")`.

Nosným bodom systému sú však dokumenty – teda inštancie procesov. Tie môžu byť voliteľne zaradené do jednej z divízií KOM alebo CNC, podľa čoho som navrhol domovskú stránku s rázcestníkom tak, ako je vidieť na obrázku 6.5. V porovnaní s návrhom 5.3 je odstránený obrázok pri pomenovaní divízie, čím sa redukovalo množstvo elementov na obrazovke. V aktuálnom systéme sa procesy typu G8D výrazne odlišujú vnútornou implementáciou, ale v novom systéme bude implementácia daná ich diagramom rovnako ako pri všetkých ostatných procesoch. Samostatná kategória pre tieto procesy teda nie je potrebná. Pod záhlavím sa nachádzajú zoznamy procesov roztriedené do divízií podľa miesta uplatnenia. Po vybratí možnosti z rázcestníka sa prechádza na pohľad so zoznamom dokumentov, pričom sa nastavuje filter na zvolený proces, čo podľa dotazníku vzišlo v otázke číslo 7 (obrázok B.5) ako väčšinová preferencia.



Obr. 6.5: Ukážka domovskej stránky nového systému Workflow.

Samotný zoznam dokumentov zobrazujem v tabuľkovom zobrazení poskytovanom balíčkom *ag-grid-vue3*, vďaka ktorému je dostupná dynamická konfigurácia stĺpcov, radenie, či filtrovanie podľa dát v stĺpci alebo v celej tabuľke. Ukážka 6.6 obsahuje tabuľku s vybranými stĺpcami podľa predchádzajúcej verzie systému Workflow v anglickej lokalizácii. Vo vrchnej časti je pomenovanie aktuálneho umiestnenia v systéme a ponuka typov procesov na zobrazenie. Nižšie sa nachádzajú možnosti rýchleho či pokročilého vyhľadávania. Vpravo je zase odkaz na založenie nového dokumentu, ktoré bolo predstavené pri obrázku 6.4. Stĺpce nie sú napĺňané vo všetkých prípadoch, nakoľko každý proces nemusí obsahovať aktivity vzťahujúce sa na všetky uvedené dátumy. Oproti návrhu z obrázku 5.2 ubudli tlačidlá záložiek. Sledovanie dokumentov podľa používateľa možno nahradiť filtrovaním tabuľky a zoznam aktivít, ktoré má používateľ vykonať, sú obsiahnuté v pohľade *Úlohy*. V spodnej časti je tiež legenda pre farebnú symboliku pri záznamoch v tabuľke – modrá značí procesy, ktoré bežia a šedá označuje dokončené procesy.

Funkcionalita, ktorá používateľom najviac chýba, je pokročilé vyhľadávanie podľa dát v dokumentoch. Nový systém ponúka túto možnosť vo vyskakovacom okne z obrázku 6.7 dostupnom na obrázku 6.6 pod tlačidlom *Advanced search*. Na demonštráciu je implementované filtrovanie podľa artiklových čísel vyskytujúcich sa v úvodnom formulári procesu, ktorý obsahuje položku s kľúčom `articleNumbers`. V okne s pokročilým vyhľadávaním možno označiť artiklové čísla, ktoré sa majú vyhľadať v úvodných formulároch a stlačením tlačidla *Filtrovať* sa aplikuje filter. Tlačidlom *Zrušiť* sa tento filter zasa odstráni a tlačidlo *Zatvoriť* zavrie toto okno. Na správne fungovanie treba zabezpečiť rovnaké pomenovanie identifikátorov vstupných polí úvodných formulárov, nakoľko vyplnené dáta sú ukladané v podobe objektov JSON, kde je kľúčom identifikátor vstupného poľa a hodnotou je vyplnený vstup od používateľa. Na rovnaký kľúč musí byť nastavený aj filter v klientskej aplikácii, aby sa podľa neho hľadalo v konkrétnej hodnote objektu JSON. Ukážková filtrácia vyhľadáva v položkách s kľúčom `articleNumbers` a možnosti na filtrovanie sú poskytované prostredníctvom triedy `externalDataRetriever`, nachádzajúcej sa na serverovej časti.

Documents

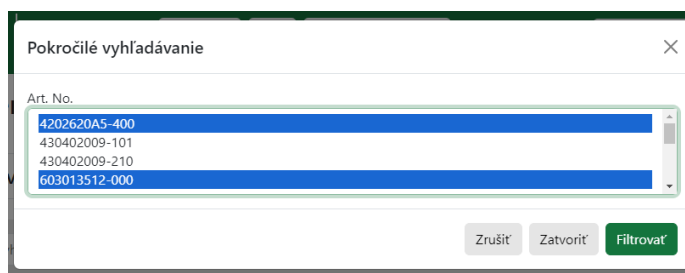
KOM - changeProposal

Quick search Advanced search + Create document

ID	Title	Submitted by	Submitted at	Resolved at	Implemented at
ZL2_0043-2024	bbb	Demo User	21. 4. 2024		
ZL2_0042-2024	aaa	Dominik Augustin	21. 4. 2024		
ZL2_0041-2024	quick openproject	Dominik Augustin	21. 4. 2024		
ZL2_0040-2024	final test 1	Dominik Augustin	21. 4. 2024		
ZL2_0039-2024	aaaaaaaaa	Demo User	15. 4. 2024		
ZL2_0038-2024	demo nazov				
ZL2_0037-2024	nnz				

State indicator ● ACTIVE ● FINISHED

Obr. 6.6: Zoznam dokumentov s anglickými popismi.



Obr. 6.7: Vyskakovacie okno pre pokročilé filtrovanie.

Dvojitým kliknutím na riadok v zozname dokumentov sa otvorí nová karta s detailom dokumentu. Tento spôsob otvárania je podľa otázky číslo 9 (obrázok 4.19) prevažujúci, na základe čoho som rovnaké správanie aplikoval pri všetkých tabuľkových zobrazeniach. Ukážka detailu procesu aj s vyplneným úvodným formulárom je na obrázku 6.8. Kliknutím na tlačidlo *BPMN diagram* sa vo vyskakovacom okne zobrazí diagram procesu, podľa ktorého otvorený dokument prebieha. Pod tlačidlom sa nachádza hlavička dokumentu zhrňujúca základné informácie o dokumente, teda jeho označenie, dátum vytvorenia, autor a názov dokumentu. Pod hlavičkou sa postupne zobrazujú formuláre, ktoré sa vyskytli v priebehu procesu. Vpravo hore je meno osoby, ktorá formulár vypracovala a čas spracovania serverom. V rovnakej úrovni vľavo je pri ďalších formulároch pomenovanie aktivity z definície procesu. Ak v inšancií procesu existujú nevypracované formuláre, zobrazia sa pod tými dokončenými, čím sa simuluje postupná tvorba dokumentu.

Administratívna časť systému ponúka správu pre počítadlá dokumentov, definície procesov a definície používateľských formulárov. Pri počítadlách sa dá nastaviť pomenovanie, príznak pre resetovanie počítadla na nový rok, voliteľný popis a taktiež počiatková hodnota, od ktorej sa má číslovať, čo sa zide pri prechode na aktuálny systém, aby sa pokračovalo v nastolenom číslovaní.

Ďalšou častou je konfigurácia definície procesu. Pohľad na zoznam definícií procesov je na obrázku 6.2. Pre proces sa dajú nastaviť vlastnosti ako názov procesu, predpona ozna-

BPMN diagram

Označenie REK_0006-2024	Vytvorené 23. 04. 2024	Vytvoril Basic User
Názov Porucha kompresora DK50 2V		

Basic User
23. 4. 2024 9:16:57

Názov dokumentu / reklamácie*

Porucha kompresora DK50 2V

Závažnosť*

Vysoká závažnosť - vyjadrenie a doplnenie info

definuj závažnosť incidentu

Zadaj výrobné číslo E1954-10-2023	Vyjadrenia* User a User b User c
--------------------------------------	-------------------------------------

Komponent

Agregat Hlava valca

vyber problematický komponent

Popis problému

Obr. 6.8: Detail založeného dokumentu.

čenia dokumentov, či zaradenie procesu do divízie alebo priradenie vytvoreného počítačľa. Vďaka voľne dostupným balíčkom od spoločnosti Camunda je možné integrovať modelovacie nástroje pre procesy a formuláre priamo do webového rozhrania, ktoré sú však plne v angličtine a neaplikujú sa na ne lokalizácie z balíčka *i18n*.

Každý formulár má svoj identifikátor, pomocou ktorého sa naň môže proces odkazovať v začiatočných udalostiach alebo aktivitách typu *User Task*. V detaile formuláru na obrázku 6.9 možno vidieť čas poslednej aktualizácie, číslo verzie a editovateľný názov. Pod ním je samotný editor formuláru so sadou základných vstupných polí v ľavej časti. V pravej časti je náhľad hotového formuláru a konfigurácia dodatočných atribútov. Pod oboma nástrojmi na nachádza pár poznámok, ktoré by mal administrátor pri práci dodržiavať, aby v systéme nevznikali chyby. Manuál k týmto nástrojom je poskytovaný v [10].

Väčšina aktivít aktuálne vykonávaných procesov workflow v spoločnosti EKOM zahŕňa vyplňanie formulárov od konkrétnych ľudí, čo vyžaduje správu povinností každého používateľa. Z tohto dôvodu sa v systéme Workflow nachádza pohľad „Úlohy“ (obrázok 6.10), kde používateľ vidí svoje povinnosti v procesoch. Tento pohľad je tvorený aktívnymi úlohami vedenými v systéme Camunda, ktoré sa získavajú pre aktuálne prihláseného používateľa. Dvojitým kliknutím na úlohu sa používateľ dostane do detailu dokumentu, kde mu je priradená nejaká úloha. Pohľad obsahuje pomenovanie aktuálneho umiestnenia v systéme, tlačidlo smerujúce na zoznam implementačných úloh vedených v systéme OpenProject a samotnú tabuľku s úlohami prihláseného používateľa. Tá obsahuje pomenovanie aktivity podľa definície procesu, dátum vytvorenia úlohy, voliteľne termín dokončenia, označenie a názov dokumentu, ku ktorému sa úlohy viažu.

Riadenie projektov nie je riešené v rámci nového systému Workflow, ako to je v stávajúcom systéme, ale využíva sa spolupráca so softvérom OpenProject, nakoľko snaha o vlastné

Používateľský formulár Doplnkové informácie

Aktualizované

22. 4. 2024 10:59:29

Verzia

1

Názov

Doplnkové informácie

The screenshot displays a user interface for configuring a form. On the left, a 'Components' sidebar offers various input types like 'Number', 'Date time', 'Text field', and 'Text area'. The central workspace is divided into 'Form Definition' and 'Form Preview'. The 'Form Definition' section shows the configuration of fields: a text input for 'Zákazník', a dropdown for 'Počet súprav', and a text area for 'Podrobnejší popis'. The 'Form Preview' section shows how these fields will appear to the user. On the right, a configuration panel for the form 'Reklamacia_doplnkove' is visible, showing the 'General' tab and the form's ID.

Obr. 6.9: Konfigurácia formuláru priamo v systéme.

Moje úlohy

Úlohy v OpenProject

Názov úlohy	Vytvorené	Termín	Označenie dokumentu	Názov dokumentu
Doplniť dodatočné informácie	23. 4. 2024		REK_0007-2024	Test proces OMP_1
Doplniť dodatočné informácie	23. 4. 2024		REK_0008-2024	DK50-10Z - porucha motora

Obr. 6.10: Zoznam procesných úloh prihláseného používateľa.

riešenie by pre firmu nebola efektívna. Spolu s projektmi sa do tohto softvéru presúvajú aj zložitejšie úlohy, ktoré priamo nesúvisia s tokom dát v procesoch workflow, ale vychádzajú z vykonaných rozhodnutí v týchto procesoch. Pre tieto úlohy sa očakáva založený projekt s názvom *Workflow*, ktorý bude vyhradený pre implementačné úlohy zo systému Workflow.

6.4 Nasadenie systému

Na uľahčenie nasadenia implementovaného systému som zvolil platformu Docker³, čo je technológia pre virtualizáciu kontajnerov podobná technológií virtuálnych zariadení (anglicky *Virtual Machines*). Zatiaľčo virtuálne zariadenia obsahujú kompletnú kópiu operačného systému, Docker komunikuje priamo s operačným systémom hostovského zariadenia s nízkou réžiou a možnosťou zdieľania vrstiev programového vybavenia naprieč kontajnermi. Využitie kontajnerizovaných aplikácií zároveň uľahčuje prenositeľnosť programov medzi zariadeniami, ktoré majú nainštalovanú platformu Docker [5].

Webový klient, server s aplikačnou logikou a behové prostredie Camunda predstavujú predpokladané zloženie kontajnerov pri nasadení do produkčnej prevádzky. Na demonštračné účely s nimi používam ešte kontajnery pre databázu MongoDB, OpenProject a server EKOM Auth. Spôsob zostavenia aplikácií je daný súbormi `Dockerfile` v priečinkoch súčastí a spustenie na platforme Docker súbormi s názvom v podobe `compose*.yaml`.

³Domovská stránka platformy Docker: <https://www.docker.com/>

Kapitola 7

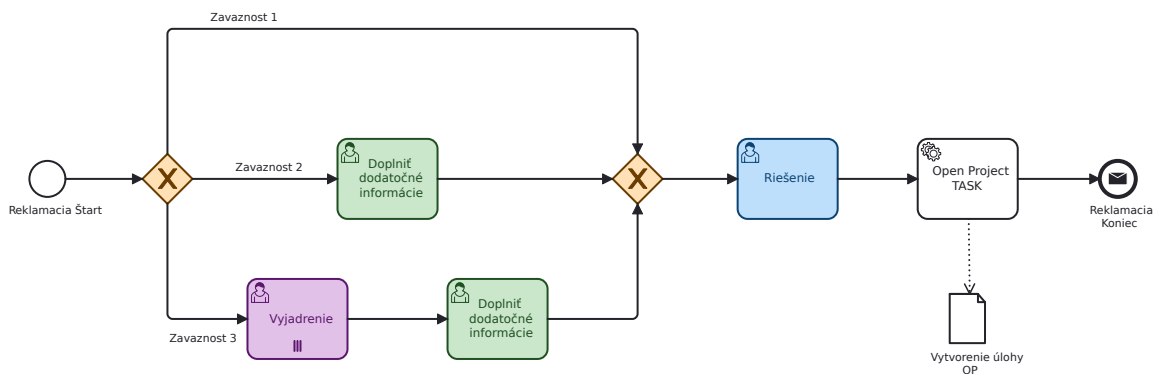
Testovanie funkčnosti a použiteľnosti nového systému

Finálne riešenie informačného systému bolo prezentované širšiemu okruhu zamestnancov, vrátane riaditeľov spoločnosti, vedúcich oddelení, ako aj vybraných rádových zamestnancov, ktorí so systémom najčastejšie pracujú. V krátkosti bol zhrnutý stav aktuálneho systému Workflow, po ktorom nasledovalo predstavenie nového riešenia a názorná demonštračná ukážka práce so systémom z pohľadu administrátora aj bežného používateľa. V závere bola otvorená diskusia s účastníkmi, ktorí sa proaktívne zaujímali o možnosti nového systému a poskytli prvú spätnú väzbu.

Viacerí na mieste prejavili záujem, aby bol systém v budúcnosti rozšírený o vedenie privátnych procesov workflow pre jednotlivé oddelenia s vlastnou správou. Používatelia by v náhlade na BPMN diagram procesu privítali farebné zvýraznenie aktuálne vykonávanej aktivity, ako aj zobrazenie pomocníka s krátkym popisom aktivity po nabehtnutí kurzorom na prvok diagramu. Tieto záležitosti však nie sú zahrnuté do požiadaviek na túto fázu vývoja. Pomocné popisy budú užitočné aj mimo náhladu BPMN diagramu, aby sa používatelia naučili rýchlejšie pracovať s novým systémom. Celkovo sa im však páčila vizuálna reprezentácia a konfigurácia procesov i formulárov, kategorizácia procesov podľa divízie či ukladanie rozpracovaných formulárov. Nasledovalo testovanie s vybranými používateľmi, ktorí si priamo skúsili prácu s novým systémom.

Ako prvý si nový systém vyskúšal jeden z administrátorov, ktorý mal za úlohu vytvoriť ukážkový model procesu. Použité nástroje na modelovanie považoval za intuitívne aj napriek tomu, že k jazyku BPMN aj k nadstavbe v podobe platformy Camunda bude na používanie potrebné naštudovanie dokumentácie. Pri prvom modelovaní potreboval viackrát poradiť a prijal by rozšírené množstvo poznámok priamo v rozhraní. Príkladom je vytvorenie procesu tak, aby sa v ňom mohli zakladať úlohy do systému OpenProject alebo využitie externého spracovávanie správ z platformy Camunda podľa položky *Topic*. Vhodným správaním by bolo priebežné ukladanie rozpracovaných definícií procesov bez nasadenia do behového prostredia Camunda. Testovacie prostredie pre odskúšanie navrhnutých procesov mu v systéme chýba. Výsledkom tejto časti je proces na obrázku 7.1, ktorý popisuje zjednodušenú formu spracovávanie reklamácie. Podľa úrovne závažnosti sa volí jedna z ciest, kde niektoré obsahujú ďalšie aktivity. Na záver sa určí riešiteľ implementačnej úlohy a vytvorí sa koreňová úloha v systéme OpenProject.

Nasledovalo testovanie koncovými používateľmi. Na tento účel bolo vybraných 5 zamestnancov, ktorí so stávajúcim systémom pracujú na dennej báze. Testovanie bolo uskutočnené



Obr. 7.1: BPMN diagram procesu vytvoreného v rámci testovania systému.

na procese reklamácie, kde bolo ich úlohou založiť nové dokumenty pre tento typ procesu, sledovať ich priebeh a reagovať na aktivity, ktoré v inštanciách nastali. Na rýchlu demonštráciu flexibility systému bol administrátorom upravený jeden z formulárov vyskytujúcich sa v procese. Táto zmena bola viditeľná, akonáhle sa proces dostal na úlohu s týmto formulárom. „Taktó vieme upravovať celé procesy v rádoch hodín namiesto dní,“ zhodnotil.

V závere som im položil pár otázok ohľadom ich prvej vlastnej skúsenosti s novým systémom. Najskôr som sa pýtal na intuitívnosť navigácie v rozhraní na stupnici od 1 (najmenej) po 10 (najviac). Štyria respondenti označili hodnotu 9 a jeden hodnotu 8. Jednoduché štýlovanie rozhrania však nevyhovovalo všetkým a starším osobám robili problém jemné farby, ktoré im splývali s bielym pozadím a nájsť potrebnú informáciu im teda trvalo dlhšie.

Použitá knižnica Bootstrap definuje paletu použitých farieb premennými, ktoré sa dajú nahradiť vlastnými hodnotami. Vhodným výberom kontrastnejších farieb bude možné uľahčiť prácu starším osobám a túto prekážku odstrániť.

Druhá otázka zisťovala mieru naplnenia očakávaní od systému. Na základe tohto testovania zamestnanci zhodnotili, že systém do značnej miery spĺňa očakávania, respektíve má na to silné predpoklady, keď sa nasadí do ostrej prevádzky.

Pri používaní systému im nič neprišlo mäťúce a ani im nič neprekvapilo, s čím by si nevedeli poradiť. Naopak, najviac sa im páčila relatívne jednoduchá vizuálna modifikovateľnosť procesov, a tým pádom flexibilita systému pri ich vykonávaní, rýchla odozva, či napojenie na systém pre projektové riadenie.

Riešenie vypracované v rámci tejto práce do značnej miery napĺňa ciele firmy EKOM, ktorá sa na základe týchto výsledkov rozhodla pokračovať vo vývoji tohto informačného systému. Hodnotenie zo strany firmy je dostupné v prílohe C.

Kapitola 8

Záver

Zámerom tejto práce bolo navrhnuť a implementovať informačný systém pre správu a riadenie procesov workflow. Špecifikom tohto zadania bolo nájsť riešenie, ktoré odstráni obmedzujúce vlastnosti existujúceho riešenia vo firme EKOM, prevádzkovaného od roku 2008.

Nový informačný systém má podobu trojvrstvovej architektúry s oddelenou databázou, vyčleneným serverom na aplikačnú logiku a samostatnou klientskou aplikáciou. Tá v prehliadači používateľa vykresľuje obsah podľa dát prijatých cez rozhranie štýlu REST, ktoré je špecifikované vo formáte OpenAPI. Klientská aj aplikačná vrstva boli napísané v jazyku TypeScript a využíva sa nerelačná databáza MongoDB. Serverová časť je určená pre behové prostredie Node.js, jej základom je aplikačný rámec Fastify a klientská časť je vytvorená v aplikačnom rámci Vue.js. Riešenie je možné nasadiť na platforme Docker pomocou poskytnutých konfiguračných súborov.

Implementácia nového informačného systému spĺňa očakávania firmy, a teda v blízkej dobe môže nahradiť doterajší systém. Nadchádzajúcou fázou je audit stávajúcich procesov s kompetentnými osobami a ich transformácia do podoby BPMN diagramov. Firma má záujem o ďalší vývoj tohto systému viacerými smermi. Systém sa môže rozšíriť o zavedenie privátnych procesov pre oddelenia, rozšírenie používateľských formulárov o vlastné elementy či validátor diagramov BPMN, respektíve ich reprezentácie v jazyku XML, ktorý by overoval dodržanie konvencií pre definovanie procesov, ktoré si stanoví firma.

Taktiež bude potrebné venovať pozornosť webovému klientovi a zapracovať spätnú väzbu z testovania na zvýšenie používateľského komfortu. K modelovacím nástrojom sa môžu vytvoriť manuály, ktoré uľahčia prácu novým i vracajúcim sa používateľom alebo pomôžu so špecifikami firemného prostredia. Do budúca bude užitočné pre každý proces definovať najdôležitejšie atribúty, podľa ktorých sa budú dynamicky meniť stĺpce v tabuľke inštancií.

Literatúra

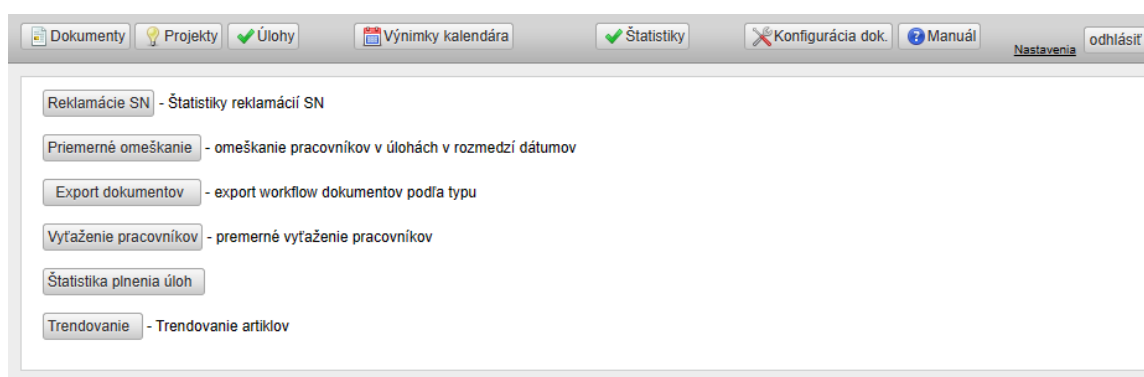
- [1] AALST, W. M. P. van der, ALDRED, L., DUMAS, M. a HOFSTEDE, A. H. M. ter. Design and Implementation of the YAWL System. In: PERSSON, A. a STIRNA, J., ed. *Advanced Information Systems Engineering* [online]. Berlin, Heidelberg: Springer, August 2004, s. 142–159. DOI: 10.1007/978-3-540-25975-6_12. ISBN 978-3-540-25975-6. Dostupné z: https://doi.org/10.1007/978-3-540-25975-6_12.
- [2] AALST, W. v. d. a HEE, K. v. *Workflow management: models, methods and systems*. Paperback. Cambridge, London: MIT Press, 2004. ISBN 0-262-72046-9.
- [3] AALST, W. v. d. a TER HOFSTEDE, A. YAWL: yet another workflow language. *Information systems* [online]. Elsevier Ltd. Jún 2005, zv. 30, č. 4, s. 245–275, [cit. 2023-10-13]. DOI: 10.1016/j.is.2004.02.002. ISSN 0306-4379. Dostupné z: <http://dx.doi.org/10.1016/j.is.2004.02.002>.
- [4] AALST, W. van der, HOFSTEDE, A. ter, KIEPUSZEWSKI, B. a BARROS, A. Workflow Patterns. *Distributed and Parallel Databases* [online]. Springer Science and Business Media LLC. Júl 2003, zv. 14, č. 1, s. 5–51, [cit. 2024-04-17]. DOI: 10.1023/A:1022883727209. ISSN 1573-7578. Dostupné z: <http://dx.doi.org/10.1023/A:1022883727209>.
- [5] ANDERSON, C. Docker [Software engineering]. *IEEE Software*. Institute of Electrical and Electronics Engineers (IEEE). Máj 2015, zv. 32, č. 3, s. 102–c3. DOI: 10.1109/ms.2015.62. ISSN 0740-7459. Dostupné z: <http://dx.doi.org/10.1109/ms.2015.62>.
- [6] ANDREATTA, R. What is the Static-Site Generation and how it works. *Medium* [online]. 13. marca 2023 [cit. 2024-01-22]. Dostupné z: <https://ferie.medium.com/what-is-the-server-side-rendering-and-how-it-works-aedffe41f8a6>.
- [7] BOOTSTRAP. *Bootstrap documentation* [online]. 2024 [cit. 2024-04-19]. Dostupné z: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>.
- [8] BRUCKNER, T. *Tvorba informačních systémů : principy, metodiky, architektury*. 1. vyd. Praha: Grada, marec 2012. Management v informační společnosti. ISBN 978-80-247-4153-6.
- [9] BUŇATA, T. *Srovnání REST, GraphQL a gRPC API v Node.js*. Praha, 2022. Diplomová práce. České vysoké učení technické v Praze, Fakulta informačních technologií.
- [10] CAMUNDA. *Camunda documentation* [online]. 2023 [cit. 2024-01-15]. Dostupné z: <https://docs.camunda.io/>.

- [11] FASTIFY. *Fastify documentation* [online]. 2024 [cit. 2024-04-17]. Dostupné z: <https://fastify.dev/docs/latest/>.
- [12] FIELDING, R. T. a TAYLOR, R. N. *Architectural Styles and the Design of Network-Based Software Architectures*. Irvine, 2000. Dizertačná práca. University of California. ISBN 0599871180.
- [13] GEORGAKOPOULOS, D., HORNICK, M. a SHETH, A. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases* [online]. 2. vyd. Springer Science and Business Media LLC. Apríl 1995, zv. 3, č. 2, s. 119–153, [cit. 2023-07-27]. DOI: 10.1007/BF01277643. ISSN 1573-7578. Dostupné z: <https://doi.org/10.1007/BF01277643>.
- [14] HRUŠKA, T. a KŘIVKA, Z. *Informační systémy (IIS,PIS) – Studijní opora* [online]. Brno: Fakulta informačních technologií VUT v Brně, február 2012 [cit. 2023-12-13]. Dostupné z: <https://www.fit.vutbr.cz/study/courses/WAP/private/podklady/Opory/OIISPojem.pdf>.
- [15] IBM CORPORATION. *IBM Security Verify Governance 10.0.0* [online]. IBM Corporation, 2021 [cit. 2023-10-08]. Dostupné z: https://www.ibm.com/docs/en/SSGHJR_10.0.0/pdf/sig-and-i-10.0.0-documentation.pdf.
- [16] JONES, M. B., BRADLEY, J. a SAKIMURA, N. *JSON Web Token (JWT)* [RFC 7519]. RFC Editor, máj 2015. DOI: 10.17487/RFC7519. Dostupné z: <https://www.rfc-editor.org/info/rfc7519>.
- [17] LAUDON, K. C. a LAUDON, J. P. *Management information systems : managing the digital firm*. 12th ed., global ed. Harlow: Pearson, 2012. ISBN 978-0-273-75453-4.
- [18] LAURENT, A. *The design of everyday APIs*. 1. vyd. Shelter Island: Manning Publications, 2019. ISBN 978-1-61729-510-2.
- [19] LEACH, P. J., SALZ, R. a MEALLING, M. H. *A Universally Unique Identifier (UUID) URN Namespace* [RFC 4122]. RFC Editor, júl 2005. DOI: 10.17487/RFC4122. Dostupné z: <https://www.rfc-editor.org/info/rfc4122>.
- [20] MILLER, D., WHITLOCK, J., GARDINER, M. a RATOVSKY, R. OpenAPI Specification v3.1.0. *OpenAPI Initiative* [online]. 15. februára 2021 [cit. 2024-04-25]. Dostupné z: <https://spec.openapis.org/oas/latest.html>.
- [21] OBJECT MANAGEMENT GROUP. *Business Process Model and Notation* [online]. 2.0.2. Object Management Group, január 2014 [cit. 2023-10-22]. Dostupné z: <https://www.omg.org/spec/BPMN>.
- [22] OPENPROJECT. *OpenProject documentation* [online]. 2023 [cit. 2024-01-16]. Dostupné z: <https://www.openproject.org/docs/>.
- [23] PIWOWARCZYK, D. Multi-tenancy vs Single-tenancy application architecture. *DEV* [online]. 20. decembra 2021 [cit. 2024-01-18]. Dostupné z: <https://dev.to/przpiw/multi-tenancy-vs-single-tenancy-4n92>.

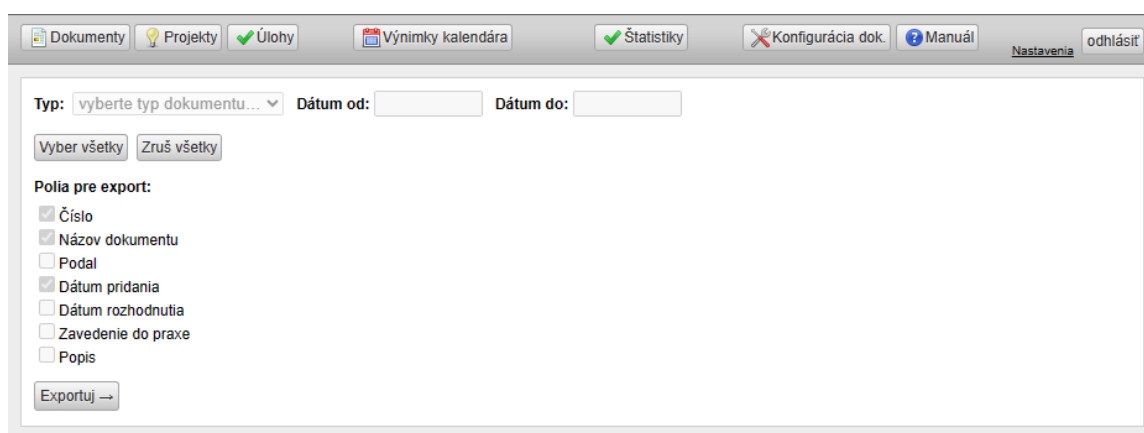
- [24] PRATT, M. K. Workflow definition. *TechTarget* [online]. Apríl 2022 [cit. 2023-07-28]. Dostupné z: <https://www.techtarget.com/searchcio/definition/workflow>.
- [25] RADGUI, M., SAIDI, R. a MOULINE, S. Design for Reuse in Business Process: Method and Experiments. *International Journal of Enterprise Information Systems* [online]. IGI Global. Október 2013, zv. 9, č. 4, s. 12–27, [cit. 2023-09-19]. DOI: 10.4018/ijeis.2013100102. ISSN 1548-1123. Dostupné z: <http://dx.doi.org/10.4018/ijeis.2013100102>.
- [26] RAJKUMAR. Software Architecture: One-Tier, Two-Tier, Three Tier, N Tier. *Software Testing Material* [online]. 2016 [cit. 2024-01-18]. Dostupné z: <https://www.softwaretestingmaterial.com/software-architecture/>.
- [27] SAAFAN, A. Why TypeScript Is the Ultimate Language for Full-Stack Development. *LinkedIn* [online]. 23. septembra 2023 [cit. 2024-04-20]. Dostupné z: <https://www.linkedin.com/pulse/why-typescript-ultimate-language-full-stack-amr-saafan>.
- [28] SHARP, A. a MCDERMOTT, P. *Workflow modeling: tools for process improvement and applications development*. 2. vyd. Boston, London: Artech House, 2009. ISBN 978-1-59693-192-3.
- [29] SODOMKA, P. a KLČOVÁ, H. *Informační systémy v podnikové praxi*. 2. aktualiz. a rozš. vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.
- [30] TAYLOR, D. Server-Side Rendering: The Pros & Cons To Consider For SEO. *Search Engine Journal* [online]. 28. marca 2023 [cit. 2024-01-22]. Dostupné z: <https://www.searchenginejournal.com/server-side-rendering/481581/>.
- [31] TAYLOR, D. Client-Side Vs. Server-Side Rendering. *Search Engine Journal* [online]. 27. apríla 2023 [cit. 2024-01-22]. Dostupné z: <https://www.searchenginejournal.com/client-side-vs-server-side/482574/>.
- [32] TAYLOR, D. Rehydration For Client-Side Or Server-Side Rendering. *Search Engine Journal* [online]. 27. júna 2023 [cit. 2024-01-22]. Dostupné z: <https://www.searchenginejournal.com/rehydration/485009/>.
- [33] THE GRAPHQL FOUNDATION. *GraphQL* [online]. 2023 [cit. 2024-01-12]. Dostupné z: <https://graphql.org/>.
- [34] VUE. *Vue.js documentation* [online]. 2024 [cit. 2024-04-19]. Dostupné z: <https://vuejs.org/guide/introduction.html>.
- [35] VÁLKA, M. *Podpora workflow na platformě JavaScript*. Brno, 2020. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií.
- [36] WESKE, M. *Business Process Management: Concepts, Languages, Architectures*. 1. vyd. Berlin, Heidelberg: Springer Nature, 2007. ISBN 978-3-540-73521-2.

Príloha A

Snímky obrazovky aktuálnej verzie systému Workflow vo firme EKOM



Obr. A.1: Ponuka modulu pre získavanie rôznych štatistík systému Workflow.



Obr. A.2: Možnosti exportu dát v zvolenom časovom intervale pre zvolený typ dokumentu a jeho polia pre export.

Názov: KOM G8D report
 Popis: G8D report medical pre kompresory
 Číselník: G8K_17-2023
 Admin skupina: G8Dkom posúdenie
 Status po ukončení dokumentu: 8 - Ukončený dokument
 Sekcie: [Pridať sekciu](#) | [Pridať vyjadrenia](#)

Názov sekcie: D0-3
 Skupina: G8Dkom D0-3 Farba: Posledná sekcia: nie Status: 2 - Návrh vo vyjadrovaní

Komponenty: D0_CNC - Príprava na proces CNC [Pridať komponentu](#)

Komponenta HEAD_GLOB Názov: Hlavicka Dokumentu 8D global Verifikácia validácia: nie ✖ ⬇ ⬆ ⬇

Komponenta D0 Názov: Príprava na proces - Kompresory Verifikácia validácia: nie ✖ ⬇ ⬆ ⬇

Tlačidlá: [Pridať tlačidlo](#)

Názov: Odoslať na posúdenie prípadu Ukončenie dokumentu: nie Prejť na sekciu: posúdenie ✖ ⬇ ⬆ ⬇

Názov sekcie: posúdenie
 Skupina: G8Dkom posúdenie Farba: Posledná sekcia: nie Status: 2 - Návrh vo vyjadrovaní

Komponenty: D0_CNC - Príprava na proces CNC [Pridať komponentu](#)

Komponenta d_dec3 Názov: Posúdenie prípadu Verifikácia validácia: nie ✖ ⬇ ⬆ ⬇

Tlačidlá: [Pridať tlačidlo](#)

Názov: Postúpiť na D4 Ukončenie dokumentu: nie Prejť na sekciu: D4 ✖ ⬇ ⬆ ⬇

Názov: Ukončiť (D8) Ukončenie dokumentu: nie Prejť na sekciu: D8 ✖ ⬇ ⬆ ⬇

Názov sekcie: D4
 Skupina: G8Dkom D4-7 Farba: Posledná sekcia: nie Status: 2 - Návrh vo vyjadrovaní

Komponenty: D0_CNC - Príprava na proces CNC [Pridať komponentu](#)

Komponenta D4 Názov: Analýza kľúčových príčin Verifikácia validácia: nie ✖ ⬇ ⬆ ⬇

Tlačidlá: [Pridať tlačidlo](#)

Názov: Postúpiť na D5 Ukončenie dokumentu: nie Prejť na sekciu: D5D6 ✖ ⬇ ⬆ ⬇

Názov: Ukončiť (D8) Ukončenie dokumentu: nie Prejť na sekciu: D8 ✖ ⬇ ⬆ ⬇

Názov sekcie: D5D6
 Skupina: G8Dkom D4-7 Farba: Posledná sekcia: nie Status: 2 - Návrh vo vyjadrovaní

Komponenty: D0_CNC - Príprava na proces CNC [Pridať komponentu](#)

Komponenta D5D6 Názov: Trvalé opatrenia;Validácia Verifikácia validácia: áno ✖ ⬇ ⬆ ⬇

Tlačidlá: [Pridať tlačidlo](#)

Názov: Postúpiť na D7 Ukončenie dokumentu: nie Prejť na sekciu: D7 ✖ ⬇ ⬆ ⬇

Názov: Ukončiť (D8) Ukončenie dokumentu: nie Prejť na sekciu: D8 ✖ ⬇ ⬆ ⬇

Názov sekcie: D7
 Skupina: G8Dkom D4-7 Farba: Posledná sekcia: nie Status: 2 - Návrh vo vyjadrovaní

Komponenty: D0_CNC - Príprava na proces CNC [Pridať komponentu](#)

Komponenta D7 Názov: Prevencia proti výskytu Verifikácia validácia: nie ✖ ⬇ ⬆ ⬇

Tlačidlá: [Pridať tlačidlo](#)

Názov: Ukončiť (D8) Ukončenie dokumentu: nie Prejť na sekciu: D8 ✖ ⬇ ⬆ ⬇

Názov sekcie: D8
 Skupina: G8Dkom posúdenie Farba: Posledná sekcia: áno Status: 3 - Čaká na rozhodnutie

Komponenty: D0_CNC - Príprava na proces CNC [Pridať komponentu](#)

Komponenta D8 Názov: Rozhodnutie Verifikácia validácia: nie ✖ ⬇ ⬆ ⬇

Tlačidlá: [Pridať tlačidlo](#)

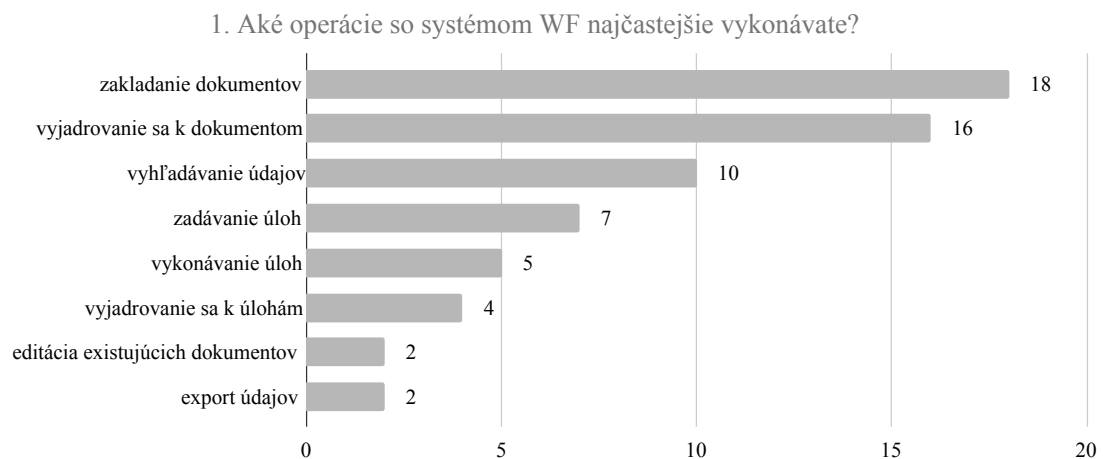
Názov: Ukončenie dokumentu Ukončenie dokumentu: áno Prejť na sekciu: ✖ ⬇ ⬆ ⬇

[Uložiť template](#)

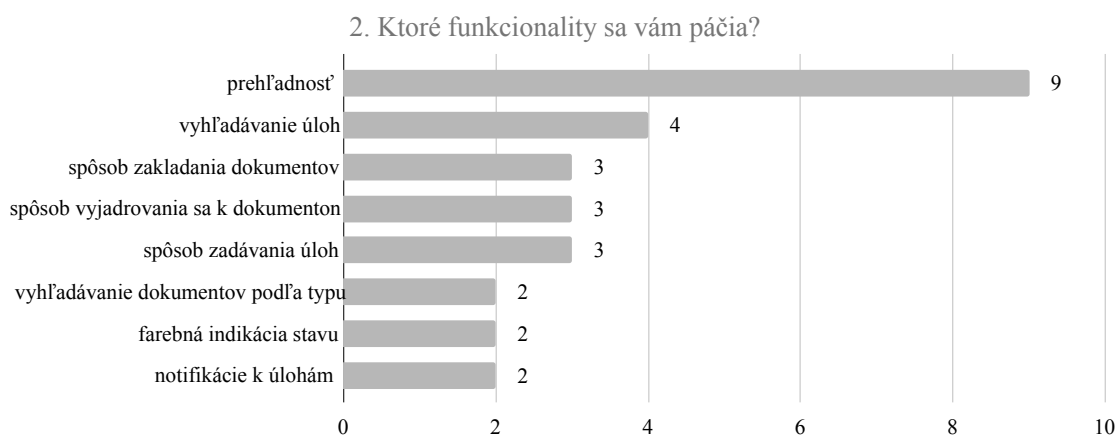
Obr. A.3: Rozšírený pohľad na konfiguráciu procesu G8D.

Príloha B

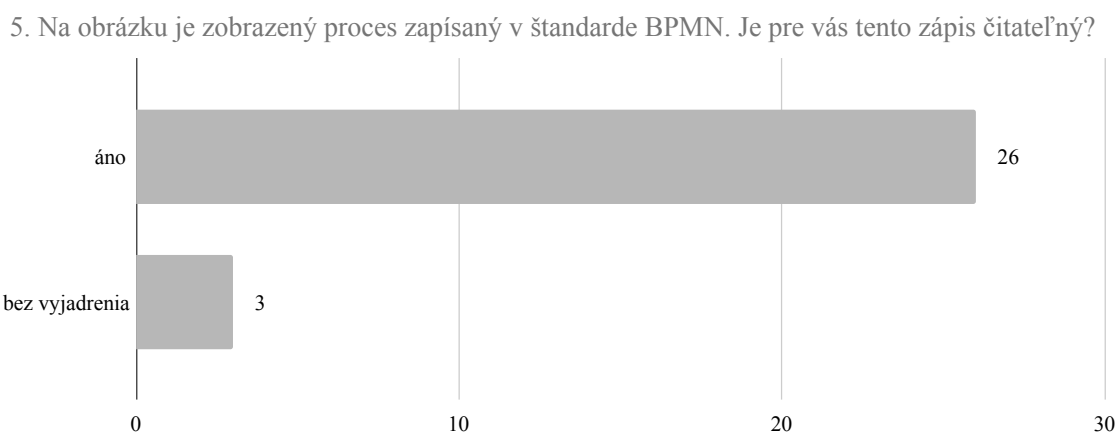
Výsledky dotazníku k aktuálnej verzii systému Workflow pre zamestnancov firmy EKOM



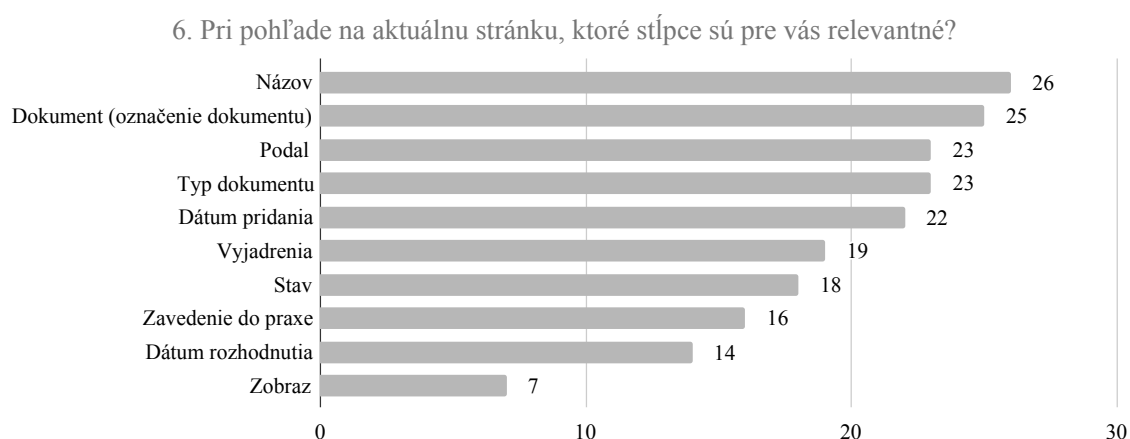
Obr. B.1: Otázka číslo 1: Aké operácie so systémom Workflow najčastejšie vykonávate?



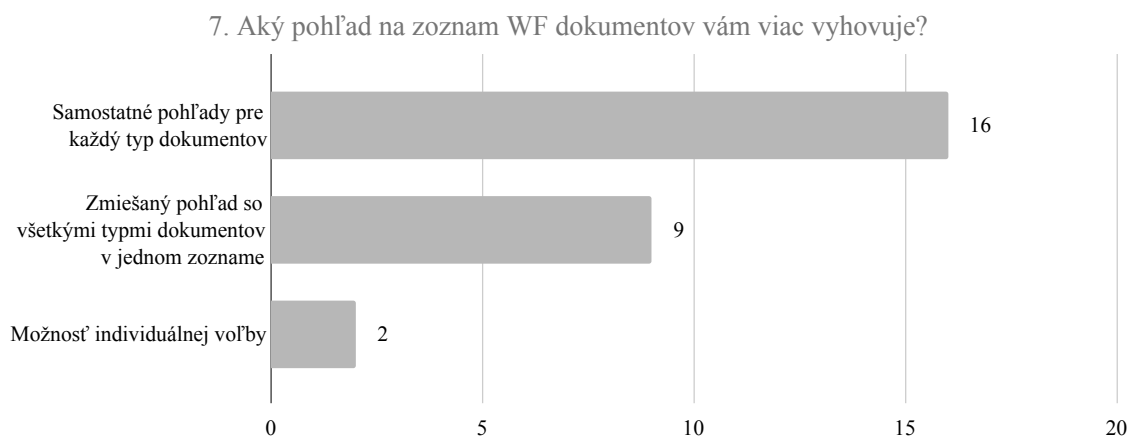
Obr. B.2: Otázka číslo 2: Ktoré funkcionality sa vám páčia?



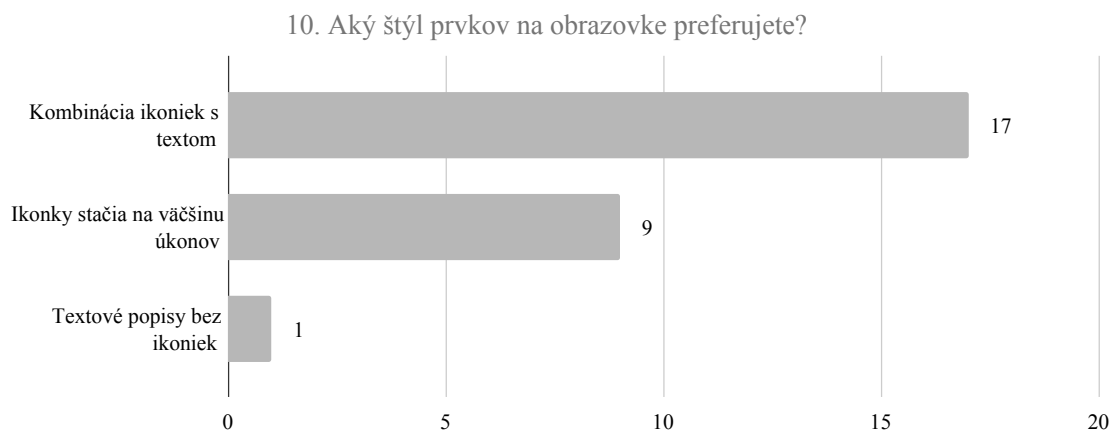
Obr. B.3: Otázka číslo 5: Na obrázku je zobrazený proces zapísaný v štandarde BPMN. Je pre vás tento zápis čitateľný?



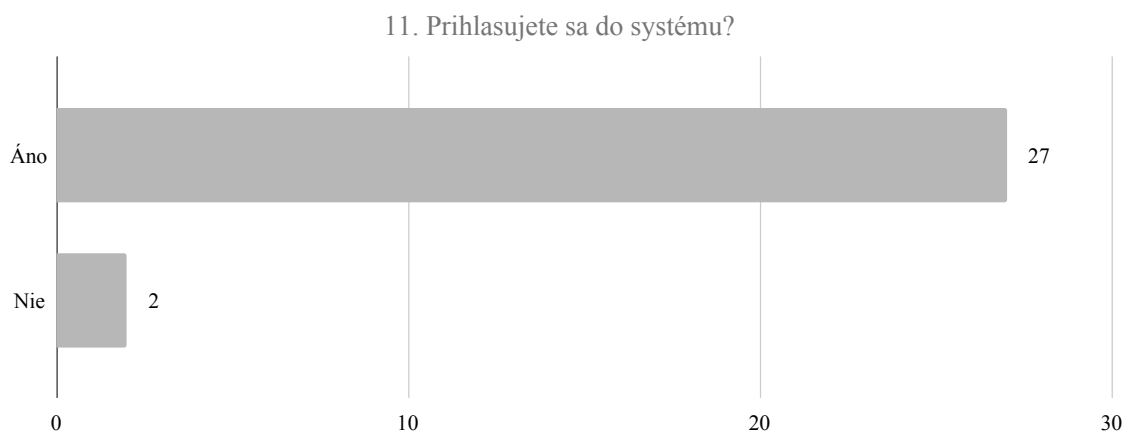
Obr. B.4: Otázka číslo 6: Pri pohľade na aktuálnu stránku, ktoré stĺpce sú pre vás relevantné?



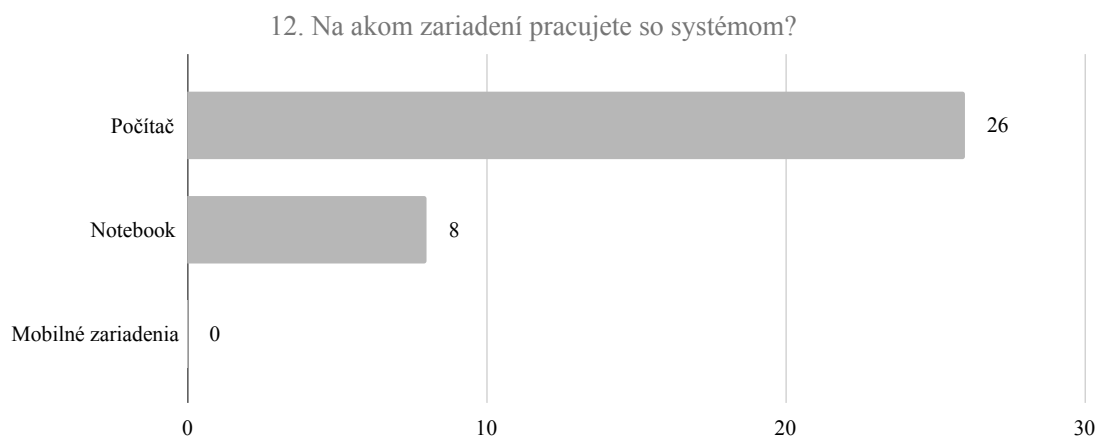
Obr. B.5: Otázka číslo 7: Aký pohľad na zoznam dokumentov workflow vám viac vyhovuje?



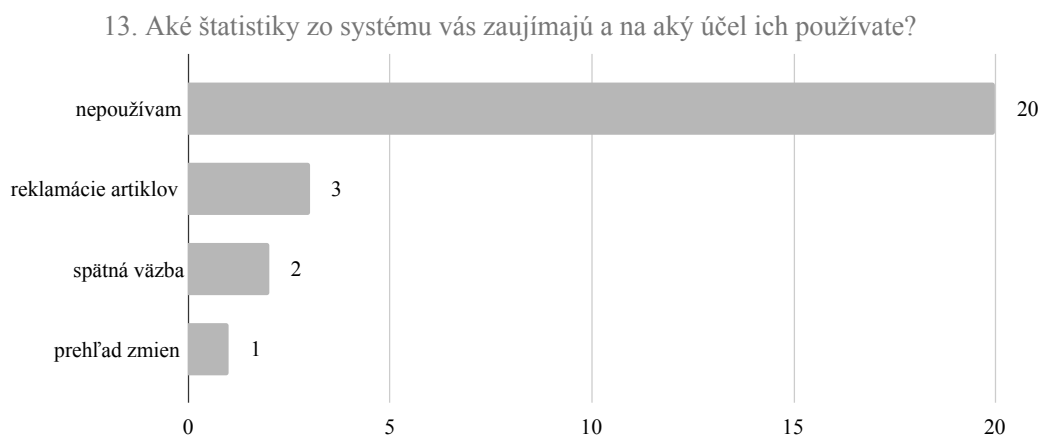
Obr. B.6: Otázka číslo 10: Aký štýl prvkov na obrazovke preferujete?



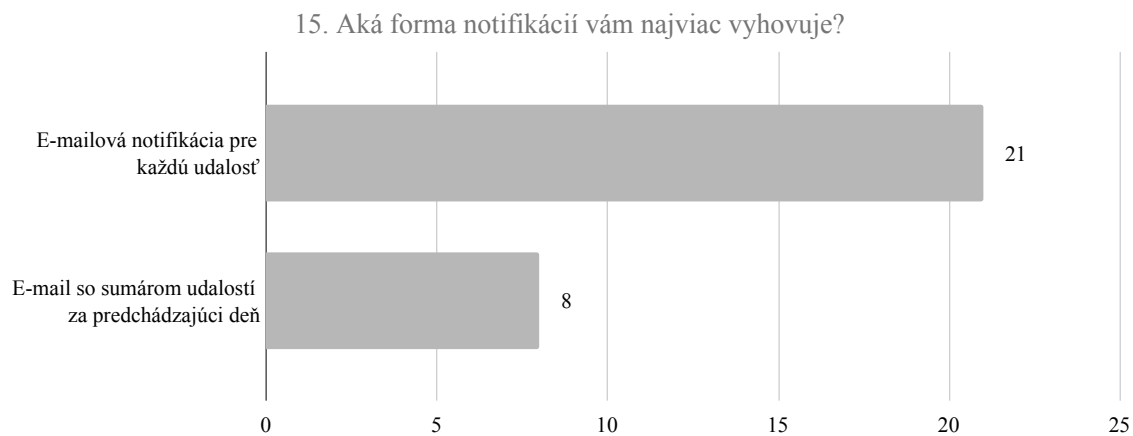
Obr. B.7: Otázka číslo 11: Prihlasujete sa do systému?



Obr. B.8: Otázka číslo 12: Na akom zariadení pracujete so systémom?



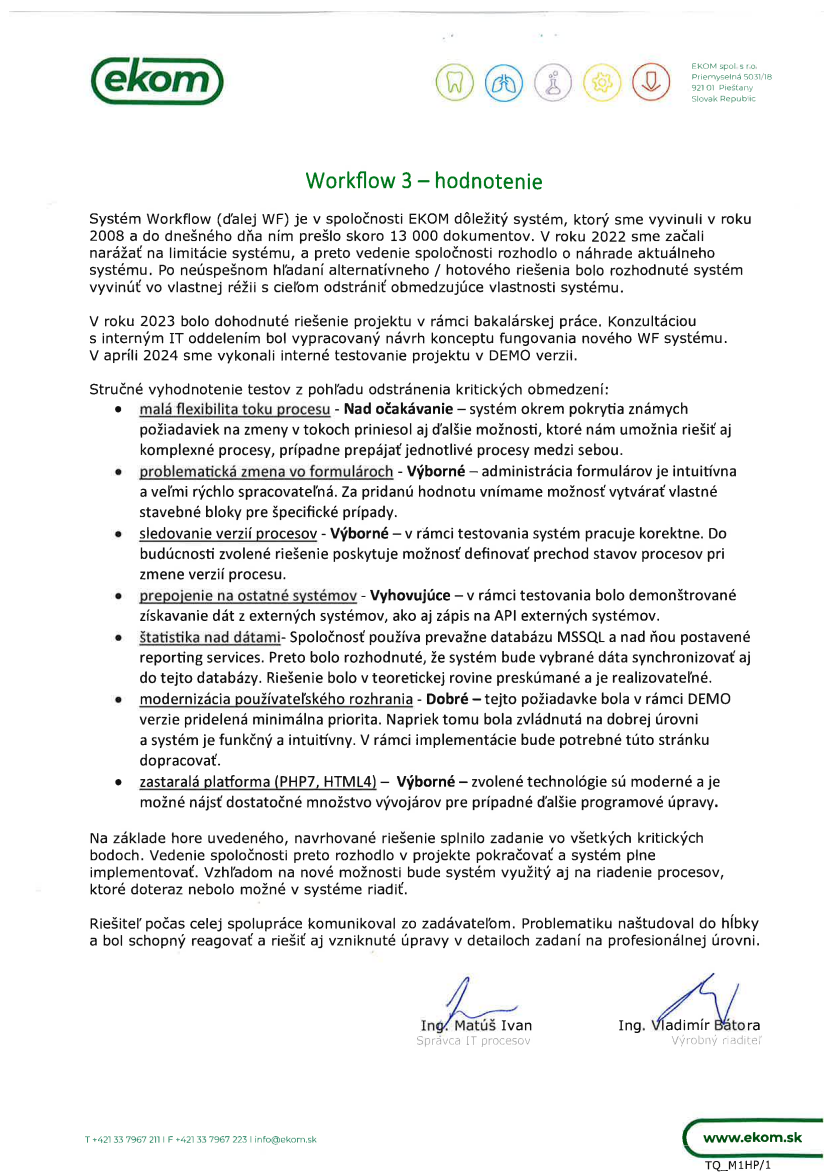
Obr. B.9: Otázka číslo 13: Aké štatistiky zo systému vás zaujímajú a na aký účel ich používate?



Obr. B.10: Otázka číslo 15: Aká forma notifikácií vám najviac vyhovuje?

Príloha C

Workflow 3 – hodnotenie



ekom

EKOM spol. s r.o.
Príemyselná 50/31/18
921 03 Prievidza
Slovak Republic

Workflow 3 – hodnotenie

Systém Workflow (ďalej WF) je v spoločnosti EKOM dôležitý systém, ktorý sme vyvinuli v roku 2008 a do dnešného dňa ním prešlo skoro 13 000 dokumentov. V roku 2022 sme začali narážať na limitácie systému, a preto vedenie spoločnosti rozhodlo o náhrade aktuálneho systému. Po neúspešnom hľadaní alternatívneho / hotového riešenia bolo rozhodnuté systém vyvinúť vo vlastnej réžii s cieľom odstrániť obmedzujúce vlastnosti systému.

V roku 2023 bolo dohodnuté riešenie projektu v rámci bakalárskej práce. Konzultáciou s interným IT oddelením bol vypracovaný návrh konceptu fungovania nového WF systému. V apríli 2024 sme vykonali interné testovanie projektu v DEMO verzii.

Stručné vyhodnotenie testov z pohľadu odstránenia kritických obmedzení:

- **malá flexibilita toku procesu - Nad očakávanie** – systém okrem pokrytia známych požiadaviek na zmeny v tokoch priniesol aj ďalšie možnosti, ktoré nám umožnia riešiť aj komplexné procesy, prípadne prepájať jednotlivé procesy medzi sebou.
- **problematická zmena vo formulároch - Výborné** – administrácia formulárov je intuitívna a veľmi rýchlo spracovateľná. Za pridanú hodnotu vnímame možnosť vytvárať vlastné stavebné bloky pre špecifické prípady.
- **sledovanie verzií procesov - Výborné** – v rámci testovania systém pracuje korektné. Do budúcnosti zvolené riešenie poskytuje možnosť definovať prechod stavov procesov pri zmene verzii procesu.
- **prepojenie na ostatné systémy - Vyhovujúce** – v rámci testovania bolo demonštrované získavanie dát z externých systémov, ako aj zápis na API externých systémov.
- **štatistika nad dátami** - Spoločnosť používa prevažne databázu MSSQL a nad ňou postavené reporting services. Preto bolo rozhodnuté, že systém bude vybrané dáta synchronizovať aj do tejto databázy. Riešenie bolo v teoretickej rovine preskúmané a je realizovateľné.
- **modernizácia používateľského rozhrania - Dobré** – tejto požiadavke bola v rámci DEMO verzie pridelená minimálna priorita. Napriek tomu bola zvládnutá na dobrej úrovni a systém je funkčný a intuitívny. V rámci implementácie bude potrebné túto stránku dopracovať.
- **zastaralá platforma (PHP7, HTML4) - Výborné** – zvolené technológie sú moderné a je možné nájsť dostatočné množstvo vývojárov pre prípadné ďalšie programové úpravy.

Na základe hore uvedeného, navrhované riešenie splnilo zadanie vo všetkých kritických bodoch. Vedenie spoločnosti preto rozhodlo v projekte pokračovať a systém plne implementovať. Vzhľadom na nové možnosti bude systém využitý aj na riadenie procesov, ktoré doteraz nebolo možné v systéme riadiť.

Riešiteľ počas celej spolupráce komunikoval zo zadávateľom. Problematiku našťudoval do hĺbky a bol schopný reagovať a riešiť aj vzniknuté úpravy v detailoch zadani na profesionálnej úrovni.

Ing. Matúš Ivan
Správca IT procesov

Ing. Vladimír Batora
Výrobný riaditeľ

T +421 33 7967 211 | F +421 33 7967 223 | info@ekom.sk

www.ekom.sk
TQ_M1HP/1

Obr. C.1: Stanovisko firmy EKOM k novému systému Workflow.

Príloha D

Obsah priloženého pamäťového média

- `xaugus14.pdf` – Táto technická správa.
- `bpSources/` – Zdrojové súbory technickej správy.
- `sources/` – Zdrojové kódy.
 - `authServerMock/` – Napodobenina serveru EKOM Auth.
 - `client/` – Klientská časť informačného systému.
 - `server/` – Serverová časť informačného systému.
 - `dockerData.zip` – Archív s demonštračnými dátami systému.