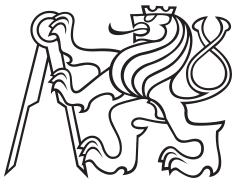
Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Centrum znalostního managementu

Reportingový nástroj pro platformu Camunda BPM

Adam Forgáč

Vedoucí: Ing. Pavel Náplava, Ph.D.
Obor: Otevřená informatika
Zaměření: Softwarové inženýrství
Květen 2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Forgáč** Jméno: **Adam** Osobní číslo: **483822**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Otevřená informatika**

Specializace: **Softwarové inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Reportingový nástroj pro platformu Camunda BPM

Název diplomové práce anglicky:

Reporting tool for Camunda BPM

Pokyny pro vypracování:

Cílem práce je vytvoření prototypu aplikace, umožňující generování reportů nad procesními daty open-source platformy Camunda BPM. Pokyny:
1) Analyzujte existující reportingové nástroje pro procesní systémy, zejména komerční produkt Camunda Optimize.
2) Představte platformu Camunda BPM a proveďte rešerši možností analýzy dat, dostupných pomocí Camunda REST API.
3) Definujte soubor požadavků na vytváření reportů z těchto dat.
4) Vypracujte technickou analýzu nové aplikace.
5) Na základě souboru požadavků navrhněte architekturu aplikace a proveďte výběr technologií k implementaci.
6) Vytvořte prototyp webové aplikace s použitím frameworku React JS, která bude umožňovat tvorbu různých typů reportů podle konfigurace v uživatelském rozhraní. Konfigurace jednotlivých reportů bude možné uložit do databáze pro pozdější využití.
7) Připravte sadu testovacích procesů a reportů. Aplikace následně podrobte uživatelskému testování.

Seznam doporučené literatury:

1. DUMAS, Marlon; ROSA, Marcello La; MENDLING, Jan; REIJERS, Hajo A. Fundamentals of business process management. Druhé vydání. Berlin, Germany: Springer Berlin, 2018. ISBN 9783662565087
2. Introducing Camunda Platform 8 [online]. USA: Camunda, 2022. Dostupné z: https://camunda.com/platform/
3. Camunda Optimize [online]. USA: Camunda, 2022. Dostupné z: https://camunda.com/platform/optimize/
4. ROLDÁN, Carlos Santana. React Cookbook: Create Dynamic Web Apps with React Using Redux, Webpack, Node. js, and GraphQL. První vydání. UK: Packt Publishing, Limited, 2018. ISBN 978-17-8398-072-7

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. Pavel Náplava, Ph.D. Centrum znalostního managementu FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **09.02.2023** Termín odevzdání diplomové práce: **26.05.2023**

Platnost zadání diplomové práce: **22.09.2024**

Ing. Pavel Náplava, Ph.D. podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	prof. Mgr. Petr Páta, Ph.D. podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

. _____
Datum převzetí zadání

_____
Podpis studenta

Poděkování

Děkuji vedoucímu mé práce
Ing. Pavlovi Náplavovi, Ph.D. za jeho profesionální přístup, ochotu a cenné rady, které mi dal při průběžných konzultacích. Dále děkuji pracovníkům Centra znalostního managementu za čas, který mi během vypracování této práce věnovali.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 16. května 2023

Adam Forgáč

Abstrakt

Cílem této práce bylo vypracovat analýzu existujících reportingových nástrojů pro BPM systémy a navrhnout, implementovat a otestovat prototyp aplikace, která bude umožňovat tvorbu reportů nad procesními daty platformy Camunda. Součástí práce byla také příprava testovacích procesů v BPMN a jejich nasazení v prostředí Camunda Platform 7. Podstatnou část práce tvořila komunikace s uživateli v rámci sběru požadavků a uživatelského testování aplikace. Dotazovaní uživatelé byli s aplikací spokojeni. Nástroj jim výrazně usnadnil analýzu procesních dat. Ve výsledku práce úspěšně splnila všechny vytyčené cíle.

Klíčová slova: Camunda, Cypress, BPM, BPMN, datová analýza, Java, JavaScript, modelování procesů, reportingový nástroj, React.js, SpringBoot

Vedoucí: Ing. Pavel Náplava, Ph.D.

Abstract

The aim of this thesis was to analyze existing reporting tools for BPM systems and to design, implement, and test a prototype application that would allow the creation of reports from process data of the Camunda platform. The work also involved preparing test processes in BPMN and deploying them in the Camunda Platform 7 environment. A significant part of the work involved communication with users for gathering requirements and user testing of the application. The surveyed users were satisfied with the application, which significantly facilitated the analysis of process data for them. In the end, the work successfully achieved all its objectives.

Keywords: Camunda, Cypress, BPM, BPMN, data analysis, Java, JavaScript, process modeling, reporting tool, React.js, SpringBoot

Title translation: Reporting tool for Camunda BPM

Obsah

Obrázky

Tabulky

Kapitola 1
Úvod

1.1 Motivace

Řízení podnikových procesů se v posledních letech stalo nedílnou součástí správného fungování mnoha společností. Jedním z populárních nástrojů v této oblasti je Camunda BPM. Ačkoli se jedná o velmi oblíbený open-source procesní software, tak bohužel neexistuje žádná volně dostupná alternativa k jeho komerčnímu analytickému nástroji Camunda Optimize.

Tento druh nástroje umožňuje vizualizovat procesní data formou reportů a poskytuje tak mocný aparát pro analytiky k monitorování a optimalizaci firemních procesů. Většina společností a institucí si nicméně nemůže dovolit tento, anebo jiný licencovaný reportingový nástroj pro zlepšení podnikových procesů. Tyto robustní nástroje mohou mimo jiné také představovat problémy při integraci do dalších systémů. Mnohdy je vyžadována pouze malá podmnožina funkcionalit, které tyto systémy nabízejí.

V rámci Centra znalostního managementu na FEL ČVUT je platforma Camunda využívána k automatizaci vybraných školních procesů. Za účelem analýzy dat a optimalizace těchto procesů jsou zaměstnanci fakulty pravidelně vytvářeny manuální reporty. Existence volně dostupného reportingového nástroje by pracovníkům tuto činnost výrazně usnadnila.

Hlavní motivací této práce je nabídnout uživatelům open-source alternativu k existujícím komerčním řešením. Tato práce zachycuje analýzu, návrh a implementaci prototypu aplikace nového analytického nástroje pro generování reportů nad procesními daty platformy Camunda.

1.2 Přínosy práce

- Usnadnění analýzy procesních dat pro uživatele platformy Camunda.
- Možnost integrace reportingového nástroje jakožto samostatné služby do jiného systému.
- Vytváření reportů bez nutnosti hlubší znalosti prostředí Camunda WebApps.

1.3 Cíle práce

- Provést analýzu existujících reportingových nástrojů pro procesní systémy a definovat možnosti pro zlepšení oproti stávajícím komerčním řešením.
- Představit procesní systém Camunda BPM a popsat způsoby získávání dat skrze Camunda REST API.
- Provést sběr požadavků na nově vznikající reportingový nástroj. Na základě dotazníku pro zaměstnance CZM definovat funkční požadavky.
- Navrhnout architekturu aplikace reportingového nástroje. Zvolit hlavní technologie pro implementaci a připravit návrhy obrazovek.
- Inicializovat projekt Camunda Platform 7 a připravit sadu jednoduchých testovacích procesů v BPMN 2.0 pro ověření funkčnosti aplikace.
- Implementovat a nasadit klientskou část aplikace.
- Otestovat aplikaci.

Část I

Seznámení s problematikou procesních systémů a reportingu

Kapitola 2
Úvod do problematiky procesních systémů

Automatizace procesů je srdcem moderního podnikového řízení. Společnosti stále častěji začínají zavádět automatizaci do svých obchodních procesů a jsou si vědomy výhod, které přináší. Lze předpokládat, že trh se softwarem pro řízení podnikových procesů čeká dlouhodobý růst. Procesní systémy usnadňují automatizaci procesů pro širokou škálu podnikových aplikací. Implementace takového systému vede ke snížení času stráveného nad opakovanými manuálními činnostmi a dlouhodobě může vést k zvýšení produktivity.

V této kapitole představuji problematiku procesních systémů a motivaci pro jejich využití v praxi. Definuji základní pojmy, se kterými budu v textu pracovat. Dále uvádím existující procesní systémy, se kterými se lze běžně setkat.

2.1 Seznámení s pojmy

V této sekci jsou stručně představeny důležité pojmy, které souvisí s tématem této diplomové práce.

- *Business process* je souhrn strukturovaných, často zřetězených, činností nebo úkolů prováděných lidmi nebo vybavením za účelem vytvoření specifické služby nebo produktu pro konkrétního uživatele nebo spotřebitele [1].
- *Business Process Management* (BPM) se zabývá dohledem na fungování organizace nebo podniku s cílem zajistit konzistentní výsledky a najít příležitosti k jejímu zlepšení. V tomto kontextu může „zlepšení“ znamenat například snížení nákladů, snížení chybovosti, a nebo také získání konkurenční výhody prostřednictvím inovace. Iniciativy ke zlepšení mohou být jednorázové nebo trvalé. BPM není o zlepšování dílčích činností, ale o řízení celých řetězců činností, událostí a rozhodnutí, které v konečném důsledku přidávají hodnotu organizaci a jejím zákazníkům. Tyto řetězce činností, událostí a rozhodnutí nazýváme procesy [2].
- *Business Process Management Software* (BPMS) pomáhá podniku definovat, nasazovat a řídit automatizované podnikové procesy. Časově náročné manuální procesy zjednodušuje na pracovní postupy v rámci

podnikových aplikací. S přidáním cloudových služeb, umělé inteligence nebo analýzy velkých dat se BPMS mění na inteligentní BPMS (iBPMS) [3].

Mezi funkcionality, které by BPMS systémy měly běžně podporovat, patří [4]:

- modelování procesů - definování činností, přechodů mezi nimi a přiřazování atributů
- automatizace procesů - nasazení, řízení a monitorování procesů
- analytické nástroje pro analýzu procesních dat

- *Životní cyklus procesu* (BPM lifecycle) se sestává z několika konsekutivních činností, které organizace používají k implementaci, správě a optimalizaci svých procesů. Tento cyklus zahrnuje tyto dílčí činnosti [5]:

1. Návrh (Design) - definice a popis stávajících procesů a jejich cílů
2. Modelování (Modeling) - vizualizace a popis procesů pomocí modelovacích nástrojů
3. Implementace (Execution) - převod navržených procesů na funkční procesy a jejich spuštění v reálném prostředí
4. Monitorování (Monitoring) - sledování a sběr dat o spuštěných procesech za účelem zjištění výkonnosti a možných nedostatků
5. Optimalizace (Optimization) - analýza dat a provedení změn, které vedou ke zlepšení výkonnosti procesů

2.2 Motivace pro použití procesních systémů

Důvodů pro zavedení BPMS existuje mnoho. Použití BPMS může být klíčové pro společnosti, které hledají způsoby, jak zlepšit jejich podnikové procesy, navýšit produktivitu nebo zlepšit kvalitu služeb. Často se může jednat o společnosti v oblasti bankovnictví, finanční společnosti nebo firmy zprostředkovávající zákaznické služby. Automatizace manuálních procesů v organizaci usnadňuje prodejním týmům generovat více potenciálních zákazníků a rychleji uzavírat více obchodů, což vede k vyšším výnosům společnosti. Mezi další časté důvody pro zavedení BPMS v podniku patří [4]:

- zvýšení efektivity díky snazšímu nasazení a řízení procesů
- snížení nákladů způsobené nižší cenou za provoz automatizovaných procesů
- zvýšení zisku zapříčiněné snížením nákladů při zachování kvality

2.3 Existující BPMS platformy

V současné době již na trhu existuje nespočet BPMS řešení. Většinu z těchto systémů lze zařadit do jedné ze dvou kategorií. Jedná se buď o komerční produkty, nebo o open-source systémy. V této sekci stručně představuji obě varianty a jejich zástupce.

2.3.1 Enterprise BPMS

Enterprise BPMS systémy jsou komerční řešení, která obvykle využívají velké společnosti se složitými podnikovými procesy a komplexními požadavky. Tato řešení mají vysoký stupeň flexibility a přizpůsobení. Nicméně vyžadují více času a úsilí na implementaci a údržbu. Typickými zástupci jsou IBM BPM nebo SAP. V této sekci je uvedeno několik populárních řešení tohoto typu. Řešení byla vybrána na základě průzkumu vykonaného společností Gartner [6].

IBM BPM

IBM Business Process Manager je BPM platforma od společnosti IBM. V rámci IBM BPM mohou být procesy modelovány, spuštěny a monitorovány. Produkt je k dispozici v lokálních i cloudových konfiguracích a podporuje mobilní zařízení. Je dostupný ve dvou verzích. IBM BPM Express je dostupnější verze, která nedisponuje pokročilejšími funkcemi a je určena spíše pro středně velké firmy. Plná verze je určena pro velké firmy, které mají obvykle několik BPM platforem a vyžadují vyšší míru automatizace [7].

SAP BPM

SAP NetWeaver Business Process Management (BPM) je jednou z komponent balíčku SAP NetWeaver od společnosti SAP. Jedná se o licencovaný komerční produkt, který umožňuje modelovat a spouštět komplexní podnikové procesy. Stejně jako většina konkurenčních produktů disponuje SAP BPM rozsáhlým uživatelským rozhraním a využívá process engine, který je v případě SAP BPM napsaný v jazyce Java [8]. Vzhledem k široké škále funkcionalit, které jsou zahrnuty v balíčku SAP NetWeaver, lze řešení řadit spíše k inteligentním BPMS (iBPMS).

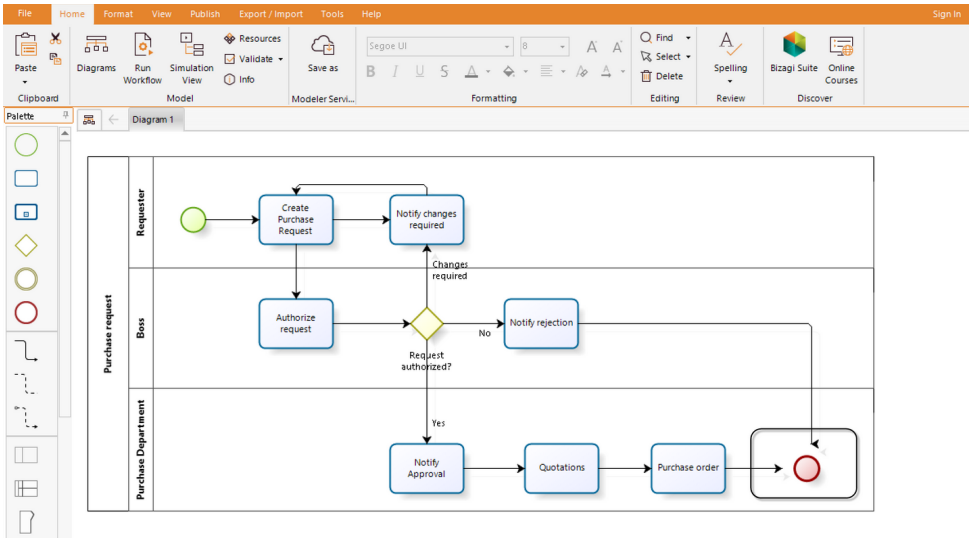
Microsoft Power Automate

Firma Microsoft má na trhu rovněž své licencované BPM produkty. Jedním z nich je Power Automate (dříve známý jako Microsoft Flow). Power Automate je platforma umožňující vytváření automatizovaných procesů s použitím malého množství kódu a snadné integrace s ostatními komponentami balíčku Office 365 od společnosti Microsoft. Řešení je dostupné v desktopové, webové i mobilní verzi [9]. Nevýhodou integrace s balíčkem Office 365 je možnost

obejití některých bezpečnostních opatření při vytváření procesů. Dochází tak ke vzniku potenciálních hrozeb.

Bizagi

Bizagi je jednou z velmi populárních low-code platforem pro automatizaci procesů. Na rozdíl od konkurence nabízí Bizagi zdarma dostupný Bizagi Modeler pro specifikaci a modelování procesů. Produkt dále zahrnuje Bizagi Studio a Bizagi Automation umožňující nasazení, monitoring a optimalizaci procesů. Modelování a nasazování procesů v Bizagi je díky přívětivému uživatelskému rozhraní a velkému množství předpřipravených konektorů velmi intuitivní. Mezi společnosti, které Bizagi v praxi používají, se řadí například Adidas nebo Audi [10].



Obrázek 2.1: Modelování procesu v Bizagi, zdroj: [10]

2.3.2 Open-source BPMS

Open-source řešení jsou alternativou ke komerčním produktům. Při jejich použití je společnost ušetřena licenčních poplatků a náklady na provoz jsou obvykle mnohonásobně nižší. Řešení s otevřeným zdrojovým kódem jsou kvůli snadné modifikovatelnosti často cílené spíše na vývojáře nežli business analytiky [11]. Níže je uveden stručný popis několika zástupců populárních open-source BPM systémů. Systémy byly zvoleny na základě průzkumu od společnosti Solutions Review [12].

Activiti

Activiti je open-source systém pro správu podnikových procesů, který je založen na standardu BPMN 2.0. Byl vytvořen v roce 2010 Tomem Baeyensem, zakladatelem projektu jBPM, a Joramem Barrezem, softwarovým vývojářem ve společnosti Alfresco. Activiti byl navržen jako lehký a flexibilní BPM

systém, který může být integrován s širokou škálou jiných softwarových systémů [13].

V roce 2013 společnost Alfresco Activiti zakoupila a stala se součástí Alfresco Digital Business Platform. Nicméně, Activiti nadále zůstává vyvíjen jako open-source projekt s komunitou uživatelů a vývojářů pracujících na jeho dalším vývoji a zdokonalování. Projekt je v současné době hostováný na platformě GitHub.

jBPM

jBPM (Java Business Process Management) je open-source framework pro správu podnikových procesů postavený na platformě Java. jBPM umožňuje uživatelům vytvářet, spouštět a spravovat podnikové procesy pomocí grafického rozhraní a podporuje standard BPMN 2.0 pro popis procesů. jBPM poskytuje funkce pro definici podnikových procesů, řízení úkolů, správu uživatelů a další. Dále jBPM vystavuje rozhraní pro integraci s jinými systémy. Lze jej integrovat do vlastních aplikací, které využívají technologie Java EE nebo SpringBoot [14].

Camunda BPM

Camunda BPM je hojně používaná open-source alternativa k robustním BPM systémům jako jsou IBM BPM nebo SAP. Umožňuje intuitivní návrh a tvorbu procesů, jejich nasazení a monitoring. Camunda se skládá z několika dílčích open-source částí, kterými jsou Camunda Modeler, Tasklist a Cockpit. Dále je také součástí platformy Camunda komerční analytický nástroj Optimize. Vše je propojeno snadno rozšířitelným Camunda Workflow Engine. Právě díky jednoduché práci s Camunda Workflow Engine a rozsáhlé dokumentaci je Camunda oblíbeným procesním systémem mezi vývojáři [15].

V následujících kapitolách budou Camunda BPM a nástroj Optimize představeny podrobněji.

2.4 Závěr kapitoly

Tato kapitola se věnovala úvodu do problematiky procesních systémů a BPM. Stručně představila pojmy, které budou v textu dále používány. Zdůvodnila výhody automatizace podnikových procesů a použití procesních systémů v praxi. Dále bylo v této kapitole provedeno rozdělení BPM platforem na komerční a open-source řešení. Pro oba typy byli stručně představeni jejich zástupci na trhu.

Další kapitoly se budou více zaměřovat na analytické nástroje pro procesní systémy. Bude do hloubky představena platforma Camunda a její analytický nastroj Camunda Optimize.

Kapitola 3

Reportingové nástroje pro procesní systémy

3.1 Úvod

Monitorování a optimalizace podnikových procesů jsou důležité fáze životního cyklu BPM. V praxi jsou běžně vykonávány business analytiky s využitím analytických nástrojů. Tyto nástroje obvykle podporují Business Intelligence (BI) funkcionality, jako může být například tvorba reportů pro analýzu spuštěných procesů [16].

V této kapitole popisuji vlastnosti reportingových nástrojů pro procesní systémy. Zmiňuji jejich přínosy a uvádím několik existujících komerčních nástrojů, které jsou běžně používány.

3.2 Klíčové vlastnosti reportingových nástrojů

V souvislosti s životním cyklem BPM (viz 2.1) se reportingové nástroje používají zejména v posledních fázích tohoto cyklu, kterými jsou monitorování a optimalizace. V těchto fázích se sleduje výkon procesů a porovnává se s očekávaným chováním. Reportingové nástroje slouží k tomu, aby umožnily získávat statistická data, vytvářet z nich reporty a tím pomoci k zlepšení efektivity procesů. Sběr dat je obvykle prováděn z různých zdrojů. Může se jednat o databáze, logovací soubory, webové služby nebo různá API.

Data jsou následně prezentována formou srozumitelných reportů tak, aby i manažeři v organizaci dokázali jejímu obsahu porozumět. Například schopnost určit délku trvání dílčích činností procesu nebo ukázat, jak může být proces vylepšen, mohou být dobré způsoby, jak zvýšit produktivitu společnosti. Bez použití reportingových nástrojů může být obecně těžké vidět přínosy ve využití BPM řešení [17].

Reportingové nástroje bývají zejména určeny pro datové analytiky, business analytiky nebo projektové manažery. Většina z nich je schopna data vykreslit pomocí tabulek, grafů a dalších způsobů vizualizace. Mezi běžně požadované vlastnosti u těchto nástrojů patří [18]:

- intuitivní uživatelské rozhraní umožňující vytváření reportů dle zvolených parametrů a metrik procesu

- možnost generování konzistentních reportů nad stejnými daty pro více uživatelů současně
- možnost uložení vybraných reportů a jejich opětovné zobrazení prostřednictvím dashboardů

3.2.1 Process Intelligence

V rámci procesních systémů má soubor vlastností požadovaných od analytických nástrojů vlastní pojmenování. Business Process Intelligence (BPI) nebo zkráceně Process Intelligence (PI) je souhrnné označení pro sběr a analýzu procesních dat s cílem identifikovat problémy v procesech a zlepšit provozní efektivitu. PI umožňuje analytikům blíže zkoumat jednotlivé kroky v procesu, dobu trvání celého procesu nebo průměrnou dobu trvání konkrétních činností. PI nástroje zahrnují [19]:

- Process mining
- Process modeling
- Task mining

Tyto činnosti umožňují na rozdíl od obecných BI funkcí cílený výběr a analýzu dat z konkrétních procesů a jejich částí.

3.3 Existující reportingové nástroje

V této sekci stručně představuji některá existující softwarová řešení pro tvorbu reportů a analýzu procesních dat. Jedná se o populární komerční produkty, které jsou v současné době na trhu. Uvedená softwarová řešení byla vybrána na základě průzkumu nejčastěji používaných analytických nástrojů a BI platforem, který byl proveden společností Gartner [20].

3.3.1 Microsoft Power BI

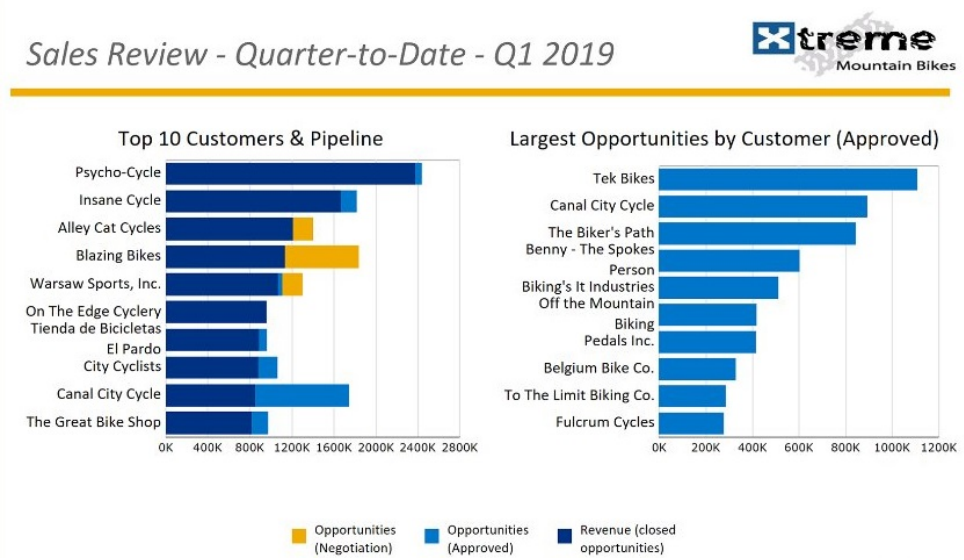
Power BI je kolekce cloudových BI služeb od společnosti Microsoft. Pomocí tabulek a dalších forem vizualizace umožňuje přeměnit nesouvisející data na smysluplné a intuitivní reporty. Power BI se skládá z desktopové aplikace Power BI Desktop, online SaaS (Software as a Service) služby a mobilní aplikace, která je kompatibilní se systémy Windows, iOS a Android [21].

Nespornou výhodou tohoto řešení je vynikající cenová dostupnost a snadná integrace s dalšími službami od společnosti Microsoft pomocí konektorů. Jednou z nevýhod Power BI je nemožnost zpracování objemnějších vstupních dat s velikostí větší než 2 GB.

3.3.2 SAP Crystal Reports

SAP Crystal Reports je podnikový reportingový nástroj využívající rozsáhlých funkcionalit SAP BI. Poprvé byl společností SAP představen v roce 1991 jako "SAP Quick Reports". Jedná se o jedno z prvních poměrně dostupných BI softwarových řešení. Produkt umožňuje společnostem sběr dat z jejich produktů a následně nad těmito daty vytvořit sofistikované reporty s podporou až 28 různých formátů, sdíleného přístupu a zobrazení v mobilních zařízeních [22].

Nevýhodou SAP Crystal Reports je poněkud matoucí licencování a nepříliš dobrá zákaznická podpora. Dále je také nepříjemné, že pro plné využití reportingových funkcionalit je nutné dokoupit další licencovaný produkt SAP Crystal Reports Server.



Obrázek 3.1: Ukázka reportu v SAP Crystal Reports, zdroj: [22]

3.3.3 Tableau

Společnost Tableau se zaměřuje na tvorbu technologií pro sběr dat a jejich vizualizaci. Její stejnojmenný produkt se stal v posledních letech jedním z nejpopulárnějších analytických nástrojů na trhu. Tableau disponuje vlastním vizuálním dotazovacím jazykem VizQL, který ulehčuje analytikům přípravu dat a práci s nimi. Řešení je v současné době dostupné ve třech variantách [23]:

- Tableau desktop - desktopová verze Tableau je ideální pro práci s lokálními daty a jejich vizualizaci
- Tableau server - enterprise verze, která je vhodná pro využití spíše ve větších firmách a organizacích
- Tableau online - nejnovější řešení od společnosti Tableau integrující funkcionality Tableau server s využitím cloudových služeb

Jedná se o vynikající analytický nástroj pro svou jednoduchost a všestrannost. Pomocí Tableau lze snadno vytvářet reporty v kombinaci s jinými BI i PI nástroji. Ačkoli Tableau nabízí výběr z mnoha druhů vizualizace dat a dovoluje také napojení na externí API, tak se nejedná o systém zcela vhodný pro práci s komplexními procesními daty. Některé druhy zobrazení, jako jsou například teplotní mapy (heatmapy) nad procesními diagramy, zde chybí.

3.3.4 Camunda Optimize

Na rozdíl od předchozích produktů nemá Optimize tak rozsáhlé zastoupení na trhu a jedná se spíše o méně známý produkt. Ve výčtu existujících reportingových nástrojů je z toho důvodu, že je to jediný existující licencovaný analytický nástroj vyvíjený přímo pro open-source platformu Camunda BPM. V rámci této práce má tedy zvláštní význam a budu jej zmiňovat i v následujících kapitolách.

Camunda Optimize je zástupcem PI analytických nástrojů. Jedná se o komerční software, který je možné integrovat přímo do Camunda BPM. Disponuje jednoduchým rozhraním, ve kterém si může uživatel nechat vygenerovat zvolené reporty pro data z procesů. Hlavní výhodou Optimize je široká nabídka možných reportů. Mezi ně patří:

- Textová pole (Labely)
- Tabulky
- Grafy
- Diagramy
- Teplotní mapy (Heatmapy)

Zejména heatmapy vytvářené z historických dat dostupných z Camunda REST API jsou pro analytiky poměrně atraktivní a spousta konkurenčních řešení je v nabídce nemá. Umožňují vizualizovat data přímo v procesním diagramu. Výhodou Optimize je také možnost definovat reporty a poté je umístit do dashboardu pro rychlý přehled [24].

Nevýhodou tohoto řešení je poměrně vysoká cena za licenci ve srovnání s ostatními PI nástroji. Dále může být problematické, že Optimize lze integrovat pouze jako modul v rámci Camunda Platform. Nelze jej tedy použít jako samostatnou aplikaci. Optimize je dostupný z webového rozhraní, ale vyžaduje od analytiků obecnější znalosti používání celé platformy Camunda BPM.

Obrázek 3.2: Ukázka dashboardu v aplikaci Camunda Optimize, zdroj: [24]

3.4 Závěr kapitoly

Tato kapitola se zaměřila na popis vlastností a využití reportingových nástrojů. Uvedla několik existujících nástrojů a jejich výhody a nevýhody. Dále představila reportingový nástroj Camunda Optimize. Některé části a funkcionality tohoto nástroje budou v rámci této práce ještě blíže představeny.

Následující kapitola bude pojednávat o celé platformě Camunda. Část se bude věnovat i integraci nástroje Optimize a využití Camunda REST API.

Kapitola 4
Camunda Platform

4.1 Úvod

Společnost Camunda se na trhu pohybuje již od roku 2008. Po dobu prvních pěti let působila především jako poradenská společnost v oblasti BPM. To se změnilo, když v roce 2013 uvedla na trh svůj nový produkt Camunda BPM, který zaznamenal velký úspěch. Po pěti letech od uvedení jej používalo přes dvě stě firem po celém světě [25]. Přestože je Camunda BPM volně dostupná open-source platforma, dá se říci, že v mnoha ohledech může konkurovat i dnešním komerčním produktům (viz kapitola 2.3). Řešení nabízí dlouhý seznam funkcionalit, které umožňují automatizaci a optimalizaci podnikových procesů.

V této kapitole je blíže představena Camunda Platform 7. Je uvedeno, z jakých dílčích částí se tento systém skládá a popsána architektura procesního enginu.

4.2 Seznámení s BPM nástroji platformy Camunda

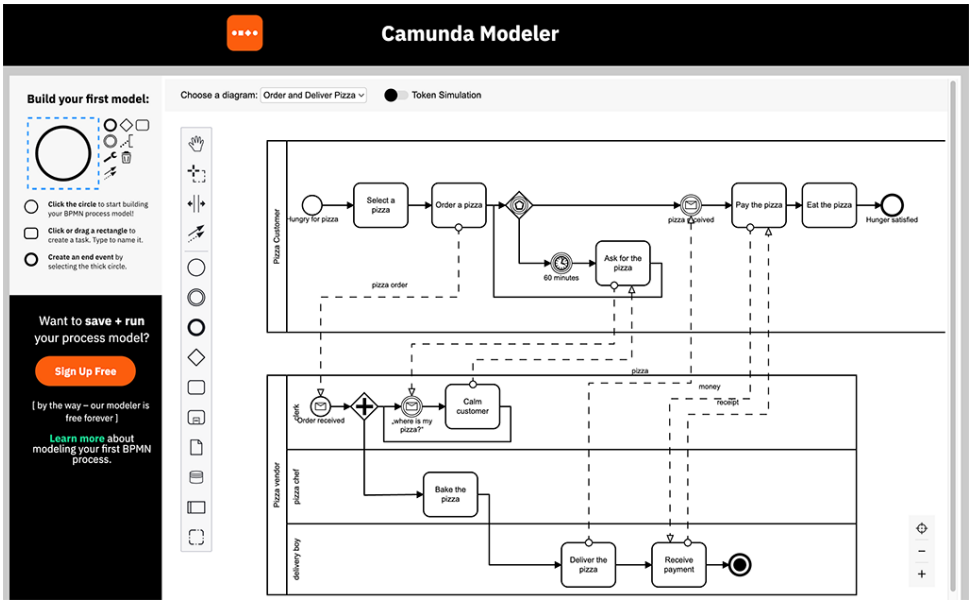
Camunda se skládá z několika BPM nástrojů, které společně umožňují návrh, automatizaci a optimalizaci procesů. V této sekci jsou tyto nástroje stručně představeny.

4.2.1 Camunda Modeler

Jedním z klíčových nástrojů každé BPM platformy je nástroj pro návrh a modelování procesů pomocí BPMN (Business Process Modeling Notation). V případě Camundy toto zajišťuje Camunda Modeler. Tento BPMN nástroj umožňuje uživatelům snadnou tvorbu procesních diagramů s použitím různých notací, jako jsou například:

- Business Process Model and Notation (BPMN)
- Decision Model and Notation (DMN)
- Case Management Model and Notation (CMMN)

Jedná se o webovou i desktopovou aplikaci, která v sobě rovněž integruje předpřipravené konektory pro napojení na různé další služby a aplikace. Díky webovému rozhraní je možné nástroj používat současně s dalšími uživateli [26].



Obrázek 4.1: Ukázka tvorby BPMN diagramu v Camunda Modeler, zdroj: [26]

4.2.2 Camunda WebApps

Camunda WebApps je souhrnné označení pro balíček nástrojů, které je možné používat prostřednictvím webového rozhraní. Na rozdíl od samostatně stahovatelného Camunda Modeler jsou všechny tyto nástroje zahrnuty již v samotné Camunda Platform 7 [15].

Tasklist

Tasklist je nástroj pro orchestraci a řízení dílčích činností procesu. Orchestrace je důležitá zejména ve chvíli, kdy některé z činností kritického podnikového procesu vyžadují manuální zpracování uživatelem. Tasklist dává možnost takové činnosti zadat přímo konkrétním uživatelům. K manuálnímu zpracování může být použit buď výchozí formulář, nebo vlastní aplikace s využitím Tasklist API. Tasklist lze také integrovat s jinými aplikacemi s použitím svého GraphQL API [15].

Cockpit

Camunda Cockpit je nástroj umožňující monitorování a analyzování běžících instancí procesu v reálném čase. Za pomoci Cockpitu je možné rychle zachytit technické problémy, které zpomalují nebo přerušují spuštěné procesy. Může se jednat o problémy, jako jsou poškozená procesní data, chybějící data,

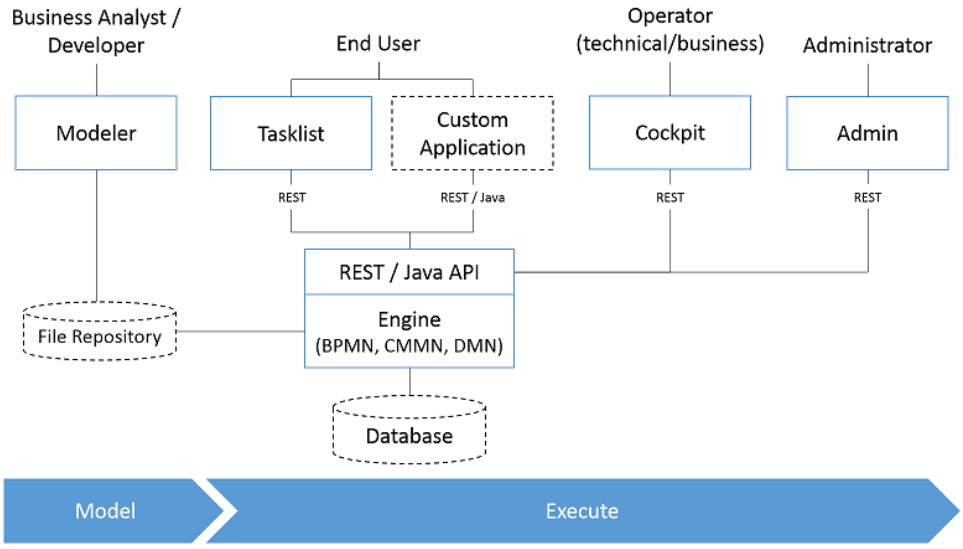
nezdařené spuštění instance procesu nebo jiná chybová hlášení. Cockpit dokáže díky přístupu k historickým datům zobrazit také všechny kroky procesu v pořadí, v jakém byly provedeny. V případě výskytu chyby lze následně opravit poškozená data, upravit logiku procesu a instanci znovu spustit [15].

Admin

Prostřednictvím aplikace Camunda Admin lze do systému přidávat uživatele a uživatelské skupiny. Uživatelům a skupinám mohou být posléze nakonfigurována přístupová práva a role pro autorizaci [15].

4.3 Architektura systému

Jádrem celého systému je Camunda Workflow Engine. Tento procesní engine zahrnuje veškerou logiku aplikace od založení procesu přes generování kroků a přenosu dat až po ukončení procesu. Existují dva hlavní způsoby provozu procesního enginu. Prvním je dříve často používaný Embedded Workflow Engine, od kterého se však v současnosti přechází k modernějšímu Remote Workflow Engine. V této sekci jsou krátce popsány oba způsoby provozu [27]. Dále je na obrázku 4.2 vyobrazena architektura Camunda Platform 7 včetně Camunda WebApps, Engine a REST API.



Obrázek 4.2: Architektura Camunda Platform 7, zdroj: [28]

4.3.1 Embedded Workflow Engine

V případě Embedded Workflow Engine je procesní engine integrován do Java aplikace formou knihovny, a je tedy přímo její součástí. V době vzniku platformy byl tento způsob provozu velmi populární. Ve většině případů se jednalo o robustní projekty, které byly nasazeny na aplikačních serverech

Jakarta EE. Od roku 2015 je možné využít i oficiální Spring Boot Starter pro založení projektu s již připraveným procesním enginem. Díky tomu je provoz v tomto režimu poměrně nenáročný a vhodný pro menší Spring Boot projekty. Nicméně v současnosti není doporučováno používat Embedded Workflow Engine u rozsáhlejších projektů. Jeho nevýhodami jsou [27]:

- podpora pouze pro Java aplikace
- žádná izolace mezi enginem a aplikací
- náročnější odstraňování problémů (Troubleshooting)
- možnost vzniku konfliktů mezi knihovnami projektu

4.3.2 Remote Workflow Engine

Jedná se o způsob provozu, při kterém je Workflow Engine oddělený od zbytku aplikace. Komunikace s ním je prováděna přes REST API. Toto řešení začalo být populární zejména s příchodem microservices. Workflow Engine je v tomto případě spravován jako samostatná služba a není závislý na programovacím jazyce ani dalších vlastnostech systému. Dovoluje také výrazně lepší škálovatelnost a je obecně doporučován jako výchozí způsob provozu pro nejnovější verzi Camunda Platform 8 [29]. Oproti embedded režimu je nevýhodou vyšší náročnost návrhu architektury systému a testování [27].

4.4 Závěr kapitoly

Tato kapitola blíže představila platformu Camunda a nástroje, ze kterých se skládá. Stručně vykreslila využití aplikace Camunda Modeler a balíčku webových aplikací Camunda WebApps. Dále uvedla rozdílné možnosti provozu Camunda Workflow Engine a jejich výhody a nevýhody.

V souvislosti s Camunda Workflow Engine je důležité připomenout, že jak již bylo uvedeno v sekci 3.3.4, Camunda vystavuje rozsáhlé veřejné API. Toto API zahrnuje i přístup k historickým datům. Jedná se o stejná data, která využívá i nástroj Optimize. Camunda tím pádem dává prostor k vytvoření vlastních alternativních řešení k jejím komerčním produktům, jako je například právě Optimize.

V následujících kapitolách se budu zaměřovat na analýzu, sběr požadavků a návrh řešení s využitím Camunda REST API v kombinaci s Camunda Workflow Engine.

Část II

Analýza nového reportingového nástroje

Kapitola 5
Analýza požadavků

5.1 Úvod

Tato kapitola zachycuje sběr požadavků na nový open-source reportingový nástroj pro Camunda Platform 7. V úvodu kapitola zkoumá požadavky na reportingové nástroje u jiných existujících řešení (viz kapitola 3). Dále se zabývá dotazníkem, který se týkal sledovaných metrik procesů v rámci Centra znalostního managementu na FEL ČVUT (CZM). Na základě dotazníku jsou následně definovány některé funkční požadavky. V další části kapitoly je poté proveden sběr požadavků na vizualizaci procesních dat.

Výsledný soubor požadavků je rozdělen na funkční a nefunkční požadavky. Požadavky obsahují jednoznačný identifikátor, název a stručný slovní popis. Soubor požadavků bude hodnotným podkladem při návrhu webové aplikace v dalších kapitolách.

5.2 Sběr informací z existujících řešení

Tato sekce pojednává o obecných vlastnostech reportingových nástrojů a funkcionalitách, kterými by měli disponovat. Dále uvádí některé specifické funkce, které jsou vhodné pro tvorbu reportů z procesních dat Camunda BPM. Při sběru informací bylo vycházeno z funkcionalit existujících nástrojů (viz kapitola 3) a dalších internetových zdrojů.

5.2.1 Vizualizace dat

Jak již bylo řečeno v předchozích kapitolách, klíčovým aspektem reportingových nástrojů pro procesní systémy je vizuální reprezentace dat. Datová vizualizace může mít velký vliv na správné rozhodování v organizaci. Je kritická zejména pro [23]:

- detekci problémů v podnikových procesech (incidenty, nezdařené úkoly)
- zjišťování rychlosti plnění podnikových procesů
- identifikaci příležitostí pro optimalizaci

Požadovaná forma a způsob zobrazení dat se mohou lišit v závislosti na konkrétním procesním systému a na sledovaných metrikách procesu. S přihlédnutím k existujícímu komerčnímu produktu Camunda Optimize (viz 3.3.4) se v případě tvorby reportů nad daty dostupnými z Camunda REST API zdají být užitečné především následující metody zobrazení:

- tabulky a labely - přímý výpis hledaných hodnot
- grafy a diagramy - zobrazení vývoje hodnot v průběhu času (např. Ganttův diagram)
- BPMN teplotní mapy (BPMN heatmap) - využití barev pro intuitivní vykreslení dat přímo do procesního diagramu

Pomocí těchto metod je nástroj schopný dát odpovědi na otázky typu: *"Kolik instancí procesu bylo spuštěno?"*, *"Které kroky procesu jsou nejčastěji navštěvovány?"* nebo *"V jakém stavu se proces nyní nachází?"*.

5.2.2 Export dat

Možnost konvertovat reporty do příslušných datových formátů je nezbytnou vlastností každého reportingového nástroje. Uložení reportu ve vhodném formátu je důležité pro následnou práci s daty a případné sdílení s dalšími analytiky nebo uživateli [30].

V tomto případě je žádoucí, aby nástroj dovedl ukládat reporty zejména ve formátu PDF (Portable Document Format). Tento formát je ideální pro uložení záznamů diagramů a teplotních map. Dále by bylo vhodné, aby nástroj umožnil export tabulkových záznamů do jednoduchého, obecně používaného formátu CSV (Comma-separated values).

5.2.3 Integrace s dalšími systémy

Většina moderních PI nástrojů využívá k sběru dat konektory, které dovolují integraci s dalšími systémy. Camunda Platform nicméně umožňuje sběr veškerých historických dat skrze Camunda REST API (viz 3.3.4). Díky tomu není požadována integrace s dalšími systémy. Pro účely sběru dat tedy postačí pouze REST API.

Dále je požadováno, aby systém nebyl integrován v rámci webového rozhraní Camunda Platform. Tak tomu je v případě Camunda Optimize. Nezávislost na Camunda WebApps a vlastní webové rozhraní dává nástroji vyšší míru kompatibility. Rovněž také neklade nároky na uživatelovu znalost prostředí Camunda WebApps.

5.3 Sledované metriky procesů v rámci CZM

CZM v současné době využívá platformu Camunda BPM pro automatizaci vybraných fakultních procesů. Ačkoli nebylo zamýšleno, aby vznikl reportingový nástroj přímo pro účely analýzy procesních dat v rámci CZM, bylo

možné získat alespoň náhled na požadované metriky procesů, které by mělo smysl sledovat.

Na základě toho vznikl dotazník, který se zabýval obecnými metrikami procesů, jenž nejsou závislé na konkrétní procesní aplikaci.

5.3.1 Vyhodnocení dotazníku

Dotazník na požadované metriky procesů byl sdílen elektronickou formou s celkem čtyřmi zaměstnanci CZM, kteří se věnují přípravě automatizovaných procesů v Camunda BPM. Výstupem dotazníku byl seznam několika metrik, které by dávalo smysl sledovat. Zmíněnými metrikami jsou:

- Doba trvání
  - celý proces
  - jednotlivé fáze procesu
  - jednotlivé činnosti / úkoly
  - fáze mezi jednotlivými stavy procesu
- Počet instancí
  - současně běžící instance
  - instance které skončili s určitým výsledkem
- Historické výsledky vyhodnocení DMN tabulky, včetně vstupních a výstupních hodnot

Přestože nebyly výstupem dotazníku samotné požadavky na reportingový nástroj, tak bylo jeho vyhodnocení velmi přínosné. Bylo zjištěno, jaká procesní data jsou požadována ke sledování. Zejména se jedná o informace o běžících instancích procesu a dob trvání dílčích částí procesu. Tyto metriky budou následně využity při tvorbě funkčních požadavků na systém.

5.4 Analýza požadavků na vizualizaci dat

Jak již bylo uvedeno v sekci 5.2.1, mezi požadované formy vizualizace dat patří label, tabulka, sloupcový graf, Ganttův diagram a heatmapa. V případě prvních třech forem se jedná o obecně používané metody zobrazení dat a není nutné je blíže představovat. U Ganttova diagramu nebo heatmapy nemusí být zřejmé, jak takový typ reportu interpretovat nebo s jakými daty pracuje. Tato sekce se zaměřuje na popis těchto dvou forem vizualizace.

5.4.1 Seznámení s teplotními mapami

BPMN teplotní mapa (BPMN heatmap) je užitečný způsob prezentace statistických dat procesu a lze se s ní setkat například u Camunda Optimize (viz 3.3.4). V této sekci je blíže vysvětleno, k čemu se teplotní mapy využívají a jaká data jsou zapotřebí k jejich vytvoření.

Teplotní mapa

Teplotní mapa je grafická reprezentace statistických dat pomocí barev. Může být využita pro různé účely. Často se lze s teplotními mapami setkat například v geografii, kde odlišné zbarvení míst na mapě představuje jisté charakteristiky. Obecně se teplotní mapy dělí na dva základní druhy [31]:

- Prostorová teplotní mapa (spatial heatmap) - vizualizace prostorově rozložených dat
- Mřížková teplotní mapa (grid heatmap) - vizualizace formou obarvených mřížek, kde dvě dimenze odpovídají dvěma typům proměnných

V kontextu této práce bude využit první z uvedených druhů teplotních map. Vysoký počet instancí splňujících dané parametry bude obarven červenou barvou a umístěn v oblasti příslušné aktivity v BPMN diagramu daného procesu. Pro nižší počet instancí se bude intenzita obarvení postupně snižovat a blížit k modré barvě.

Sledované metriky procesů pro tvorbu teplotních map

Při vytváření reportu bude uživateli umožněno si zvolit typ zobrazení. Tento typ zobrazení říká, která metrika procesu bude zvolena při vykreslení heatmapy. Vzhledem k sledovaným metrikám procesů v CZM (viz 5.3) by měl být umožněn výběr z následujících druhů zobrazení:

- Počet instancí - teplotní mapa znázorňující celkový počet instancí pro aktivity vybraného procesu
- Celková doba trvání - teplotní mapa ukazuje sumu dob trvání jednotlivých aktivit, které jsou nebo byly spuštěny
- Průměrná doba trvání - teplotní mapa ukazuje aritmetický průměr trvání aktivit, které jsou nebo byly spuštěny

Pro všechny druhy zobrazení platí, že statistická data jsou brána napříč všemi instancemi procesu. Vždy se jedná pouze o instance poslední nasazené verze zvoleného procesu. Uživatel by měl mít rovněž možnost výběru stavu hledaných instancí.

5.4.2 Ganttův diagram

Ganttův diagram je grafický nástroj, který zobrazuje časovou osu a seznam úkolů pro vybraný proces. Na diagramu bývá vykresleno, kdy začíná a kdy končí každý úkol. Diagram je užitečný pro plánování projektů a umožňuje snadno získat přehled o stavu projektu, identifikovat problémy a plánovat další kroky [32].

Analýza dat pro tvorbu Ganttova diagramu

V případě tvorby Ganttova diagramu bývá obvykle požadováno, aby byly zobrazeny jak dokončené, tak i nadcházející úkoly. Často bývají součástí diagramu také přechody mezi úkoly. Orientace přechodů i časy úkolů mohou být definovány dynamicky podle statistických dat, anebo staticky na základě expertního odhadu analytika [33].

Vzhledem k nedostatku statistických dat pro rozumnou predikci dob trvání nadcházejících úkolů budou využita historická data procesu pouze pro vykreslení časů již dokončených úkolů. Data pro nadcházející úkoly budou definována vlastními proměnnými v BPMN souboru. Dále nebude požadováno, aby byly přechody součástí diagramu. Jejich vykreslení by mohlo být značně komplikované, protože soubor může obsahovat i přechody zahrnující takové druhy aktivit, které by neměli být součástí Ganttova diagramu.

5.5 Funkční požadavky

Funkční požadavky říkají, co bude systém uživateli umožňovat. Jedná se o důležitý podklad při vývoji a návrhu aplikace. Funkční požadavky mohou dále také specifikovat nároky na způsob vyhledávání, reportování nebo přístup k datům [34].

V této sekci jsou zpracovány funkční požadavky na nový reportingový nástroj. Každý požadavek má vlastní název a popis. Pro lepší přehled je také označen unikátním identifikátorem. Požadavky byly zformulovány na základě obecných vlastností reportingových nástrojů (viz sekce 5.2), sledovaných metrik procesů v rámci CZM (viz sekce 5.3) a požadavků na tvorbu teplotních map (viz sekce 5.4.1). Výsledný soubor požadavků je uveden v následující tabulce.

ID	Název	Popis
FR01	Table report	Systém umožní uživateli vytvořit report procesních dat na základě konfigurace v UI ve formě tabulky.
FR02	Count report	Systém umožní uživateli vytvořit report počtu spuštěných instancí procesu/aktivity na základě konfigurace ve formě textu.
FR03	Heatmap report	Systém umožní uživateli vytvořit BPMN heatmap report s možností výběru zobrazení počtu nebo doby trvání jednotlivých aktivit.
FR04	Incident report	Systém umožní uživateli vytvořit report statistik obsahujících informace o incidentech procesů ve formě grafu.
FR05	Gantt report	Systém umožní uživateli vytvořit report ve formě Ganttova diagramu.
FR06	Filtrace výsledků	Systém umožní uživateli filtrovat nalezené výsledky v tabulkovém reportu dle řetězce.

FR07	Stránkování výsledků	Systém umožní uživateli stránkování tabulkového reportu a navigaci mezi stránkami.
FR08	Zobrazení proměnných procesu	Systém umožní uživateli zobrazit proměnné procesních instancí v tabulkovém reportu.
FR09	Výběr dle proměnných procesu	Systém umožní uživateli tvorbu reportu dle názvu a hodnot proměnných procesu.
FR10	Export do CSV	Systém umožní uživateli exportovat report ve formě tabulky do formátu CSV.
FR11	Export do PDF	Systém umožní uživateli exportovat report ve formě heatmapy či grafu do formátu PDF.
FR12	Uložení konfigurace	Systém umožní uživateli uložení konfigurace reportu.
FR13	Zobrazení konfigurace	Systém umožní uživateli vygenerovat nový report z uložené konfigurace.
FR14	Smazání konfigurace	Systém umožní uživateli odstranění konfigurace reportu.
FR15	Přehled konfigurací	Systém umožní uživateli zobrazit uložené konfigurace reportu.
FR16	Editace konfigurace	Systém umožní uživateli upravit uložené konfigurace reportu.
FR17	Nový report	Systém umožní uživateli přejít do sekce tvorba nového reportu.
FR18	Přihlášení	Systém umožní uživateli přihlásit se do systému prostřednictvím emailu a hesla.
FR19	Odhlášení	Systém umožní uživateli odhlásit se.
FR20	Nový uživatel	Systém umožní přidání nového uživatele.

Tabulka 5.1: Funkční požadavky

5.6 Nefunkční požadavky

Nefunkční požadavky se na rozdíl od funkčních požadavků nevztahují přímo k funkcím systému. Kladou spíše nároky na vlastnosti systému, jako jsou například použitelnost, dostupnost nebo zabezpečení. Mohou také specifikovat standardy kvality kladené na proces vývoje [35].

V následující tabulce je uveden soubor nefunkčních požadavků na nový reportingový nástroj.

ID	Název	Popis
NFR01	Uživatelská přívětivost	Systém uživateli umožní snadnou navigaci mezi sekcemi.
NFR02	Kompatibilita	Systém bude kompatibilní s prohlížeči Google Chrome (ve verzi 111.0.5 a novější) a Mozilla Firefox (ve verzi 111.0.1 a novější).
NFR03	Nezávislost na Camunda WebApps	Systém nebude závislý na webovém rozhraní Camunda WebApps.

Tabulka 5.2: Nefunkční požadavky

5.7 Závěr kapitoly

Kapitola popisovala sběr požadavků na nově vytvářený reportingový nástroj pro platformu Camunda. Nejdříve byl proveden výčet obecně požadovaných vlastností u nástroje tohoto typu. Výchozími zdroji informací byly existující komerční produkty, zejména Camunda Optimize.

Dále se kapitola věnovala dotazníku na sledované metriky procesů v rámci Centra znalostního managementu na FEL ČVUT a analýze požadavků na vizualizaci dat. Výstupy dotazníku byly využity při tvorbě souboru požadavků. Soubor byl rozdělen na část funkčních a část nefunkčních požadavků. Tyto požadavky budou potřebnými stavebními kameny v následujících kapitolách o návrhu aplikace.

Kapitola 6

Technická analýza

6.1 Úvod

Po vytvoření souboru požadavků na nový systém nastává čas na přípravu technického zadání. Nejdříve budou pomocí metody FURPS zmapovány požadavky na funkčnost, použitelnost, spolehlivost, výkon a podporovatelnost. Poté bude vytvořen technický návrh v podobě diagramu tříd. Diagram bude popisovat entity definované na základě požadovaných funkcí systému a vztahy mezi nimi.

6.2 FURPS analýza

FURPS je zkrácený výraz označující kategorie, do kterých lze rozdělit požadavky na systém. Byl vytvořen společností Hewlett-Packard. Později byl rozšířen o znaménko "plus", které reprezentuje dodatečné požadavky na implementaci, návrh nebo právní aspekty. Níže uvedený seznam shrnuje kategorie FURPS [36]:

- Functionality (funkčnost) - Definuje obecné požadavky na to, co by měl systém umět. Může se jednat o požadavky na rozhraní, bezpečnost, apod.
- Usability (použitelnost) - Popisuje, jak by měl systém vypadat a jaký dojem by měl mít na uživatele. Jedná se například o požadavky na vzhled, ovládání či responzivitu aplikace.
- Reliability (spolehlivost) - Určuje, jak by měl být systém spolehlivý. Tím může být myšleno to, s jakou pravděpodobností bude systém nedostupný, nebo jak často dochází k systémovým chybám.
- Performance (výkon) - Tyto požadavky popisují rychlost systému, náročnost na paměť, databázi, atd.
- Supportability (podporovatelnost) - Jedná se o požadavky na údržbu, flexibilitu a testování aplikace.

V následujících podsekcích je provedeno rozdělení požadavků na nový reportingový nástroj do dílčích kategorií.

6.2.1 Funkčnost

Požadavky na funkčnost systému již byly definovány v tabulce 5.1. Níže jsou tyto požadavky přerozděleny do logických sekcí, které nepřímo souvisí s UI nově vytvářené aplikace. Později bude tato organizace požadavků využita při návrhu obrazovek a usnadní orientaci mezi požadovanými funkcionalitami systému. Jako reference na existující funkční požadavky jsou využity identifikátory a názvy z tabulky.

Hlavní nabídka

- FR15 - Přehled uložených konfigurací
- FR17 - Nový report
- FR18 - Přihlášení
- FR19 - Odhlášení

Tvorba nového reportu

- FR01 - Table report
- FR02 - Count report
- FR03 - Heatmap report
- FR04 - Incident report
- FR05 - Gantt report
- FR06 - Filtrace výsledků
- FR07 - Stránkování výsledků
- FR08 - Zobrazení proměnných procesu
- FR09 - Výběr dle proměnných procesu
- FR10 - Export do CSV
- FR11 - Export do PDF
- FR12 - Uložení konfigurace

Přehled uložených konfigurací reportů

- FR13 - Zobrazení konfigurace
- FR14 - Smazání konfigurace
- FR16 - Editace konfigurace

6.2.2 Použitelnost

1. Nová aplikace bude vytvářena formou single-page application (SPA).
2. Novou aplikaci bude možné používat v prohlížečích Google Chrome a Mozilla Firefox.
3. Navigace mezi sekcemi bude zajištěna prostřednictvím navigačního panelu.
4. Veškerá konfigurace generovaného reportu bude nastavitelná uživatelem v rámci jednoho formuláře.
5. Použitelnost aplikace bude ověřena uživatelským testováním prototypu aplikace.

6.2.3 Spolehlivost

1. Spolehlivost aplikace bude ověřena prostřednictvím uživatelského testování.
2. Pro ověření spolehlivosti aplikace budou vytvořeny end-to-end testy.

6.2.4 Výkon

1. Aplikace bude schopna efektivně zpracovávat velké množství dat (řádově několik stovek instancí procesu) a generovat reporty v krátkém čase (řádově v desítkách milisekund na požadavek).
2. Aplikace nebude projevovat degradaci výkonu při nárůstu počtu uživatelů.

6.2.5 Podporovatelnost

1. Aplikace bude podporovat anglickou lokalizaci.
2. Aplikace bude obsahovat návod ke spuštění na vlastním zařízení. Návod bude obsažen v souboru README v kořenovém adresáři projektu.

6.3 Diagram tříd

Na základě funkčních požadavků bylo možné definovat entity, které se v aplikaci budou vyskytovat. Entity a vztahy mezi nimi jsou zachyceny v diagramu tříd (viz Příloha B). Vzhledem k tomu, o jaký druh aplikace se jedná, bylo namodelováno jen několik tříd. Diagram je tvořen z tříd uživatele, konfigurace reportu a přidružených atributů. Atributy konfigurace reportu zahrnují všechny parametry potřebné k sestavení požadavku na Camunda REST API. Umožňují tedy opakované vygenerování reportu pro danou uživatelskou konfiguraci.

6.4 Závěr kapitoly

V první části se kapitola zaměřila na zmapování technických požadavků na novou aplikaci s použitím FURPS analýzy. Požadavky na funkčnost byly dále rozděleny do logických sekcí, od kterých se bude v pozdějších kapitolách odvíjet návrh UI. V druhé části byl představen diagram tříd, který zachycuje entity relevantní pro návrh systému.

Kapitola 7
Analýza Camunda REST API

7.1 Úvod

V rámci předchozí kapitoly byly zmapovány požadavky na funkčnost nově vytvářené aplikace. Vybrané funkcionality, které se vztahují zejména k tvorbě nového reportu, budou vyžadovat přístup k datům aktuálně spuštěných procesních instancí. Tato sekce ověřuje, zda lze veškerá potřebná data získat pomocí Camunda REST API.

7.2 Přehled dat dostupných z Camunda REST API

Jak již bylo uvedeno v kapitolách 3 a 4, Camunda vystavuje rozsáhlé REST API skrze které lze získat veškerá historická data o spuštěných procesech a dílčích činnostech těchto procesů. Vzhledem k tomu, že i komerční řešení Camunda Optimize (viz 3.3.4) využívá k získávání dat právě toto REST API, lze předpokládat, že tento zdroj dat bude pro tvorbu reportů dostatečný.

V této sekci je provedeno zmapování požadovaných dat a REST API endpointů, z kterých by mělo být možné data získat. Výchozím zdrojem informací byla v tomto případě dokumentace Camunda BPM ve verzi 7 [28].

7.2.1 Informace o instancích procesu

Lze předpokládat, že u většiny reportů nás budou zajímat informace o spuštěných instancích procesu. Níže jsou uvedeny některé z endpointů, které tato data vystavují. Výčet zahrnuje příklady volání HTTP (Hypertext Transfer Protocol) metod, kterými lze data získat. Výsledky těchto volání budou moci být využity při implementaci reportů pokrývající funkční požadavky FR01 (zobrazení procesních dat v tabulce) a FR02 (zobrazení počtu instancí).

- `GET /history/process-instance/count` - toto volání vrací počet spuštěných instancí procesu
- `GET /history/process-instance` - volání vrací pole objektů, které obsahují základní informace o instancích procesu a zahrnují:

- id - identifikátor instance procesu
- startTime - začátek procesu (ve formátu yyyy-MM-dd'T'HH:mm:ss)
- endTime - konec procesu (ve formátu yyyy-MM-dd'T'HH:mm:ss)
- durationInMillis - doba trvání procesu v milisekundách
- startUserId - id uživatele, který spustil proces
- state - stav instance (např. ACTIVE, COMPLETED, SUSPENDED...)

Obě volání lze provést také s doplňujícími parametry. Metody následně vrátí pouze takové výsledky, které splňují podmínky těchto parametrů. Mezi doplňující parametry patří například:

- processInstanceId - zúžení výsledků podle id instancí
- processDefinitionId - filtrace dle identifikátoru definice procesu
- processDefinitionName - filtrace dle názvu procesu
- finished - pouze dokončené instance procesu
- unfinished - pouze nedokončené instance procesu
- sortBy - řazení na základě kritéria
- startedBy - pouze instance spuštěné daným uživatelem
- variables - filtrace dle hodnot vybraných proměnných procesu
- startedAfter / startedBefore / finishedAfter / finishedBefore - filtrace dle data spuštění nebo dokončení

7.2.2 Data pro generování heatmap

Jedním z reportů, který by měla vytvářená aplikace být schopna nabídnout je heatmapa. K jejímu vykreslení je zapotřebí XML soubor zachycující daný proces a soubor historických dat o aktivitách všech instancí procesu. Camunda REST API oba tyto požadavky splňuje:

- GET /history/process-definition/id/statistics - volání vrací statistická data zahrnující počet běžících, zrušených či dokončených instancí daného procesu
- GET /process-definition/id/xml - volání vrací BPMN model procesu ve formátu XML

7.2.3 Informace o aktivitách

Pro tvorbu reportů zobrazujících detailnější informace o aktivitách procesu bude možné opět využít Camunda History REST API. Analogicky k 7.2.1 lze s použitím parametrů získat historická data o aktivitám a celkovém počtu instancí aktivit:

- `GET /history/activity-instance/count` - volání vrací počet spuštěných instancí aktivit v rámci všech běžících procesů (lze filtrovat pouze pro daný proces pomocí parametru `processInstanceId`)
- `GET /history/activity-instance` - volání vrací pole objektů, které obsahují informace o aktivitách. Na rozdíl od procesních instancí zahrnují data navíc:
  - `processInstanceId` - id instance procesu ke kterému aktivita náleží
  - `assignee` - uživatel kterému je aktivita přidělena

7.2.4 Informace o incidentech

Další často sledovanou metrikou u business procesů je počet incidentů pro daný proces (souvisí s FR04). Incidenty obvykle představují nějaké události, které indikují problém při spuštění procesní instance. Camunda REST API umožňuje získat statistická data o incidentech:

- `GET /process-definition/statistics` - vrací statistická data procesu seskupená podle definic procesu. Data obsahují:
  - `incidents` - pole objektů reprezentující incidenty
  - `failedJobs` - celkový počet nezdařených úkolů pro běžící instance procesů

7.2.5 Data pro generování Ganttova diagramu

K zobrazení informací o již hotových úkolech je možné využít data o aktivitách (viz 7.2.3), které zahrnují čas spuštění i ukončení jednotlivých aktivit. Pro vizualizaci informací o nadcházejících úkolech je ideální mít definované vlastní atributy pro všechny úkoly v XML souboru. Tyto atributy slouží k predikci doby trvání úkolů a jsou jimi:

- `camunda:dueDate` - nejzazší termín dokončení (interval v ISO 8601 standardu)
- `camunda:expectedDuration` - očekávaná doba trvání úkolu (interval v ISO 8601 standardu)

7.3 Závěr kapitoly

V této kapitole bylo ověřeno, že Camunda REST API bude dostačujícím zdrojem informací o běžících procesech pro nově vytvářený reportingový nástroj. Byla zmapována všechna potřebná volání a popsány návratové hodnoty, ze kterých se budou později vytvářet reporty. Následující kapitoly se budou zaměřovat na popis architektury nově vytvářené aplikace a přípravu návrhů obrazovek.

Kapitola 8

Architektura aplikace a výběr technologií

8.1 Úvod

Tato kapitola se zaměřuje na návrh architektury nově vytvářené aplikace a výběr technologií. Bude provedeno základní rozdělení technologií použitých v rámci frontendové a backendové části řešení. Zvolené technologie budou později využity v rámci implementace. V závěru kapitoly je představen diagram komponent, který zachycuje dílčí části aplikace a jejich propojení.

8.2 Architektura aplikace

Požadavky na využití Camunda REST API pro účely získávání a zpracování dat v klientské části aplikace jsou značně vymezující z hlediska architektury. Dále jsou zde požadavky na ukládání dat konfigurací reportů do databáze, což vyžaduje komunikaci s další službou. Nejeví se tedy jako dobré řešení volit například monolitickou architekturu. Může se jednat o dobrý návrh pro malé projekty, ale s rostoucími požadavky na funkcionalitu dochází k horšímu škálování. Z uvedených požadavků vyplývá, že dobrou volbou by mohlo být využití servisní nebo client-server architektury. Oba návrhy umožňují škálování frontendové a backendové části aplikace. Servisní architektura navíc umožňuje implementaci samostatných komponent s oddělenými funkcionalitami a tím výrazně zlepšuje celkovou škálovatelnost [37].

V kontextu této práce se dá na architekturu aplikace nahlížet jako na kombinaci client-server a servisní architektury. Serverová část řešení je reprezentována samostatnou službou Camunda Platform 7 a klientskou aplikací je nově vytvářená aplikace reportingovového nástroje. V principu lze ale brát klientskou část jako samostatnou službu. Výslednou aplikaci by tak mělo být možné integrovat i do dalších systémů s například servisní architekturou a jejichž součástí je služba poskytující Camunda REST API.

8.3 Backend

V předchozí kapitole bylo ověřeno, že Camunda REST API poskytuje veškerá procesní data, která budou zapotřebí při tvorbě reportů v klientské části

aplikace (viz 7). Backendovou částí vyvíjené aplikace bude tedy Camunda BPM. Další komponentou z pohledu backendvé části řešení bude databáze pro ukládání dat o reportech.

8.3.1 Camunda Workflow Engine

Jádrem aplikace Camunda BPM je její Workflow Engine, který může být provozován buď jako Java serverová aplikace nebo samostatná služba (viz 4.3). Z hlediska architektury je možné využití obou dvou způsobů provozu. Embedded Workflow Engine dává však větší prostor pro customizaci a umožňuje pohodlnější lokální vývoj. Pro účely této práce a vývoje aplikace bude tedy využita tato forma provozu. Nicméně použití Remote Workflow Engine v rámci případné integrace aplikace do většího systému by nemělo být překážkou. Obě formy provozu umožňují přístup ke Camunda REST API.

8.3.2 Databáze pro ukládání reportů

Další potřebnou komponentou aplikace je databáze, která poslouží pro účely ukládání konfigurací reportů. Struktura těchto dat není předem dána, neboť se bude jednat o přesný opis konfigurace reportu v rámci uživatelského rozhraní. Převážně by se jednalo o parametry požadavků na Camunda REST API.

V tomto ohledu není vhodné volit SQL databáze, jako je například PostgreSQL nebo MySQL. Tato řešení jsou vhodná pro ukládání strukturovaných dat s pevně daným schématem. Pro ukládání nestrukturovaných dat s různými parametry se jeví jako lepší řešení dokumentově orientované NoSQL databáze. Ideálním kandidátem je Google Firestore. Jedná se o zástupce dokumentově orientovaných NoSQL databází. Umožňuje ukládat data ve formě dokumentů, které jsou uspořádány do kolekcí [38].

Firestore je navíc součástí Google Cloud Platform a nabízí tudíž snadnou integraci s dalšími Google Cloud službami. Práce s ním je snadná a přístup k datům je možný přímo z frontendu bez nutnosti implementace serverové části. Jedná se o ideální řešení pro ukládání konfigurací reportů.

8.4 Frontend

Při implementaci klientských aplikací se lze běžně setkat s technologiemi, jako jsou například HTML, JavaScript nebo CSS. Pro tvorbu aplikace se rovněž nabízí použití některého ze známých JavaScriptových frameworků. V této sekci je uveden stručný popis několika vybraných frameworků. U každého z nich jsou zmíněny jeho výhody a nevýhody. V závěru sekce je proveden výběr technologie.

8.4.1 Angular

Angular je populární JavaScriptový framework založený na komponentách. Umožňuje vytvářet multiplatformní aplikace. Při práci s frameworkem Angular

lze často narazit na pojmy, jako jsou TypeScript, Transpiler nebo Web Component. Naučit se pracovat s tímto frameworkem muže být náročné i pro zkušeného vývojáře. Křivka učení není tak strmá jako u jiných JavaScriptových technologií [39].

Jedná se o framework vhodný k tvorbě rozsáhlých webových aplikací. Pro účel tvorby aplikace menšího rozsahu se nejeví jako vhodné řešení.

8.4.2 Vue.js

Stejně jako Angular i Vue vychází z komponent. Vue je oproti němu však mnohem snazší na naučení a práce s ním je uživatelsky přívětivější. Umožňuje tvorbu aplikací s použitím Vue Command Line Interface nebo vložením skriptu do HTML souboru. Vue je díky své výborné škálovatelnosti ideálním řešením pro vytváření rozsáhlejších klientských aplikací [40].

Nevýhodou Vue.js je jeho omezená podpora. Framework byl vyvíjen poměrně malou skupinou vývojářů a ačkoli se jedná o velmi populární framework, tak není jeho dokumentace a podpora dobrá jako u konkurenčních JS frameworků.

8.4.3 React.js

React.js je JavaScriptová knihovna, která je spravována společností Meta (dříve Facebook) a na jejím vývoji se podílí komunita několika dalších firem a vývojářů. Umožňuje vyrábět pokročilá uživatelská rozhraní za využití opakovaně použitelných komponent. React je možné použít pro vývoj klientských i serverových aplikací. V porovnání s frameworky Angular a Vue má React výrazně lepší učící křivku [41].

Vzhledem k jednoduchosti a popularitě React.js se jedná o velmi dobrou volbu pro projekty menšího rozsahu. Pro uživatele, kteří nejsou příliš obeznámeni s tvorbou klientských aplikací existuje mnoho podpůrných materiálů a bohatá uživatelská dokumentace.

8.4.4 Ember.js

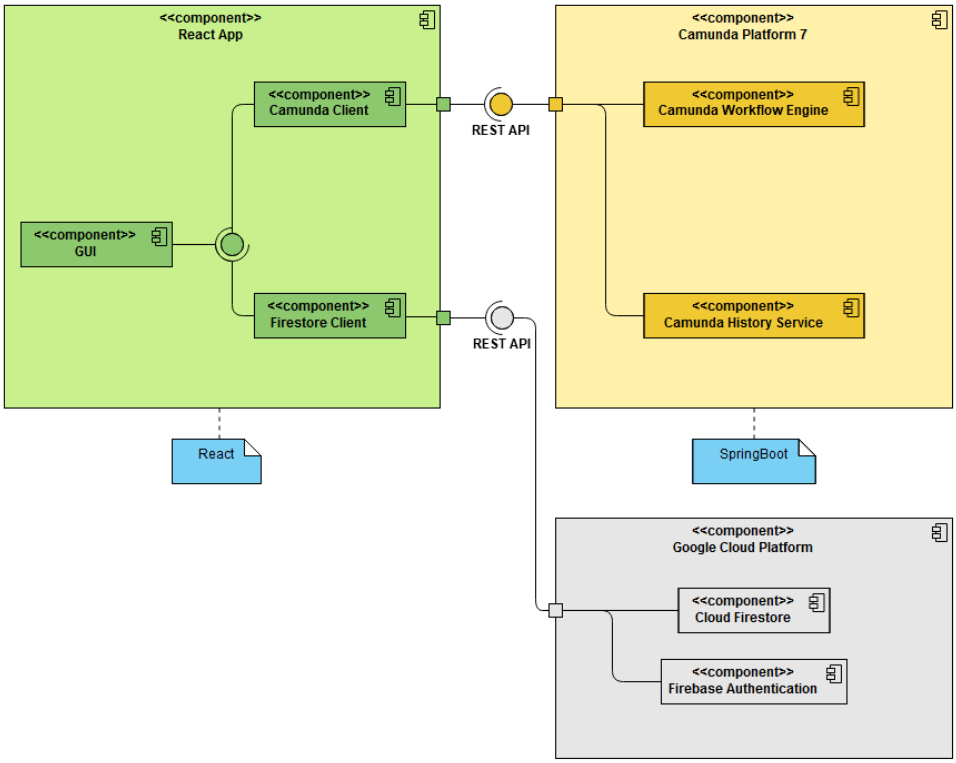
Ember.js patří od roku 2015 k populárním open-sourcovým JavaScriptovým frameworkům. Umožňuje tvorbu dynamických webových aplikací, které mohou obsahovat prvky, jako jsou například dashboardy, komunikační kanály nebo uživatelská fóra [42]. Aplikace Netflix, LinkedId, Nordstrom a mnohé další používají tento framework.

8.4.5 Shrnutí a výběr technologie

Většina z uvedených frameworků je založená na komponentách a obecně se dá říci, že každý z nich poskytuje dostatek funkcionalit pro implementaci klientské aplikace. Nicméně s přihlédnutím ke snadnému využití a rozsáhlé dokumentaci byla zvolena pro účely této práce knihovna React.

8.5 Diagram komponent

Na základě informací z předešlých sekcí bylo možné identifikovat klíčové části aplikace a popsat komunikaci mezi nimi. Ve výsledku se jedná pouze o tři hlavní komponenty, které spolu komunikují výhradně pomocí REST API. Níže je vyobrazen diagram, který tyto komponenty a komunikaci mezi nimi zachycuje.



Obrázek 8.1: Diagram komponent

8.6 Závěr kapitoly

Kapitola popsala návrh architektury nové aplikace. Dle návrhu by se mělo jednat o servisní architekturu. Toto řešení by mělo umožňovat snadnou integraci samotné klientské aplikace (reportingového nástroje) do dalších systémů využívajících procesní engine Camunda.

Dále byl proveden výběr technologií pro backendovou a frontendovou část řešení. Backend by se měl sestávat z Embedded Camunda Workflow Engine, který zahrnuje technologie Java a SpringBoot. Frontendová část bude implementována s využitím populárního frameworku React. V závěru kapitoly byl představen diagram komponent, který zmíněné komponenty a komunikaci mezi nimi popisuje.

Kapitola 9
Návrh UX/UI

9.1 Úvod

V předchozích kapitolách bylo popsáno, že nově vznikající aplikace bude uživateli umožňovat tvorbu reportů na základě konfigurace parametrů volání Camunda REST API. Následně bude možné tyto konfigurace uložit do přehledu a dále s nimi pracovat. Tato kapitola stručně uvádí požadavky na grafický návrh klientské aplikace a nastiňuje podobu a vizualizaci jednotlivých reportů. Dále jsou představeny návrhy obrazovek, které budou využity při implementaci aplikace.

9.2 Požadavky na UX/UI

Převážná většina požadavků na grafický návrh aplikace vychází z kapitol *Analýza požadavků* 5 a *Technická analýza* 6. Webová aplikace by se měla sestávat ze dvou výchozích sekcí a přihlašovacího okna. Přecházení mezi sekcemi by mělo být možné za pomoci navigačního panelu.

9.2.1 Tvorba nového reportu

V této sekci se vyskytuje formulář, který obsahuje vstupy odpovídající parametrům Camunda REST API požadavků. Dále se zde vyskytuje prostor pro vygenerovaný report a dvě formulářová tlačítka umožňující export a uložení konfigurace. Níže jsou uvedeny stručné popisy UX/UI pro všechny požadované druhy reportu.

Count report

Počet instancí aktivit či procesů odpovídajících daným parametrům je vyobrazen formou jednoduchého textového popisku. Jelikož export do CSV/PDF pro tento druh reportu nemá žádný praktický význam, nebude tato možnost ani nabízena.

Table report

V případě tabulkového reportu se pod vyhledávacím formulářem zobrazuje tabulka s výsledky. Každý řádek tabulky reprezentuje jednu instanci procesu nebo aktivity. Hodnoty ve sloupcích odpovídají množině dat, kterou vrací odpovídající volání REST API (viz 7.2.1). U procesních instancí jsou ve výsledcích zahrnuty i proměnné procesu. Detail proměnných je možné zobrazit v samostatném modálním okně. Nad a pod tabulkou se dále nachází ovládací prvky pro výběr počtu řádků na stránku, navigaci mezi stránkami a aktivaci filtrování.

Heatmap report

Jak již bylo zmíněno, heatmap report je v rámci nově vznikající aplikace vizualizován formou prostorové teplotní mapy (viz 5.4.1). Z hlediska návrhu UX/UI to znamená, že v oblasti určené pro report bude zobrazen BPMN diagram zvoleného procesu a do něj vykreslena příslušná teplotní mapa. Pro lepší přehled by mělo být umožněno zobrazit konkrétní metriky (např. počet instancí, doba trvání) při najetí myši nad obarvenou oblast. Dále bude možné zobrazit níže pod reportem soubor všech statistických dat použitých k tvorbě heatmapy a barevnou legendu, která slouží k interpretaci hodnot teplotní mapy.

Incident report

Statistická data z REST API mohou obsahovat pole incidentů nebo celkový počet nezdařených úkolů pro všechny nasazené procesy (viz 7.2.4). V UI by měl mít uživatel na výběr, která metrika ho při tvorbě reportu zajímá. Následně je zobrazen sloupcový graf. Na vodorovné ose se nachází všechny nasazené procesy včetně verzí procesu. Na svislé ose jsou počty instancí. Ke každé definici procesu jsou přidruženy sloupce reprezentující:

- celkový počet všech běžících instancí procesu
- počet nezdařených instancí
- počet incidentů (samostatný sloupec pro každý nalezený druh incidentu)

Gantt report

Ganttův diagram se skládá z horizontální osy, která zobrazuje časovou osu procesu, a vertikální osy, která obsahuje seznam úkolů a aktivit potřebných k dokončení procesu. Na diagramu jsou jednotlivé úkoly zobrazeny jako proužky, které se mohou překrývat, aby ukázaly souvislost mezi úkoly a časovými požadavky. Každý proužek zobrazuje odhadovaný časový interval, který bude potřebný k dokončení daného úkolu.

V tomto případě je zapotřebí brát v potaz, zda proces vůbec obsahuje všechny potřebné atributy k jeho zobrazení. Pro definice procesu, které

neobsahují atributy `dueDate` a `expectedDuration` nebude diagram možné vytvořit a uživateli bude zobrazena chybová hláška. V ostatních případech bude vypočtena predikovaná doba trvání nadcházejících úkolů na základě níže uvedeného postupu:

- Začátek úkolu:

$$start = processStartDate + dueDate - expectedDuration$$

- Konec úkolu:

$$end = processStartDate + dueDate$$

kde:
processStartDate = datum spuštění procesu
dueDate = doba po které by měl být úkol již hotov
expectedDuration = očekávaná doba trvání úkolu

9.2.2 Přehled konfigurací

Přehled konfigurací zobrazuje tabulku se všemi reporty, které si uživatel uložil. Pro pohodlnější vyhledávání by mělo být možné výsledky filtrovat dle názvu. Každý ze záznamů je možné upravit, zobrazit nebo přejmenovat.

9.3 Návrhy obrazovek

Příprava návrhů obrazovek je důležitá pro prvotní nastínění vizuální podoby aplikace. Návrh byl zaměřen na tvorbu low-fidelity prototypu klientské aplikace za pomoci jednoduchého internetového nástroje Moqups[43]. V prototypu je demonstrováno rozložení komponent na obrazovce. Nejsou zde řešeny detaily týkající se designu UI.

Návrhy obrazovek byly v průběhu jejich tvorby konzultovány s pracovníky CZM. Při konzultacích byly k návrhům vzneseny připomínky týkající se především nedostatečné intuitivnosti uživatelského rozhraní v sekci "Tvorba nového reportu". Veškeré připomínky byly do návrhů zapracovány. Ukázky návrhů obrazovek lze najít v příloze C.

9.4 Závěr kapitoly

Kapitola shrnula požadavky na grafický návrh aplikace. Pro každou z hlavních sekcí webové aplikace byly uvedeny požadavky na UI komponenty, které by měla obsahovat. Dále byly vytvořeny návrhy obrazovek ve formě low-fidelity prototypu, které byly průběžně konzultovány se zaměstnanci CZM. Tyto návrhy bude možné využít při implementaci frontendové části aplikace.

Část III

Tvorba prototypu aplikace

Kapitola 10

Realizace aplikace

10.1 Úvod

Tato část práce začíná popisem implementace prototypu nové aplikace. Obecně slouží prototyp jako základní funkční verze aplikace, která umožňuje ověřit správnost a úplnost návrhu. V kontextu této práce je nahlíženo na vyvíjený prototyp, jako na aplikaci, jenž integruje všechny funkcionality definované v rámci technické analýzy (viz 6). Aplikace nicméně nepokrývá zcela všechny možné kombinace získaných procesních dat a jejich zobrazení formou reportu. V tomto ohledu je aplikace volně rozšiřitelná o další druhy zobrazení dat.

Implementace zahrnuje přípravu backendové služby Camunda Platform 7 a tvorbu klientské části aplikace s využitím frameworku React. Součástí práce je rovněž i kontejnerizace a nasazení backendové služby v prostředí Google Cloud Platform. Předlohou pro vývoj aplikace byly především ukázky návrhů obrazovek (viz Příloha C), soubor funkčních požadavků (viz 5.1) a diagram tříd (viz Příloha B).

10.2 Inicializace Camunda Platform 7

V rámci přípravy backendové služby Camunda BPM byl vytvořen Maven projekt pomocí webového nástroje Camunda Automation Platform 7 Initializr [44]. Tento nástroj umožňuje snadné vytvoření SpringBoot služby Camunda Platform 7 včetně specifikace modulů, které mají být součástí projektu. Pro účely této práce byly zahrnuty moduly:

- REST API (viz 7)
- Webapps (viz 4.2.2)
- Spin (knihovna pro zpracování XML a JSON souborů)

Implementace služby byla následně rozšířena o konfiguraci přístupu k REST API, jenž umožňuje lokální vývoj. Na závěr byly do projektu přidány také BPMN soubory testovacích procesů a soubor Dockerfile. Testovací procesy jsou automaticky nasazeny po spuštění SpringBoot aplikace. Pro persistenci nasazených procesů využívá aplikace lokální H2 databázi.

10.3 Implementace klientské aplikace

V této sekci jsou stručně popsány dílčí části vývoje klientské aplikace reportingového nástroje. Jsou uvedeny některé z použitých technologií a knihoven pro tvorbu reportů.

10.3.1 Vývojové prostředí

K vývoji klientské aplikace bylo využito populární JavaScriptové IDE WebStorm [45] od společnosti JetBrains. Jedná se o pokročilé vývojové prostředí, které nabízí funkcionality jako jsou našeptávač, kontrola kvality kódu nebo vlastní terminál. Byť se jedná o komerční produkt, Fakulta elektrotechnická ČVUT poskytuje svým studentům bezplatnou licenci.

Dále byla pro práci s databází a nastavování přístupových práv do aplikace využita konzole Google Cloud Firebase, která je přístupná prostřednictvím webového rozhraní. Tato konzole rovněž umožňuje správu autentizace nebo přidání nových uživatelů do aplikace.

10.3.2 Založení projektu

React aplikace vychází z projektu Create React App [46], který usnadňuje nastavení projektu a jeho adresářové struktury. Do projektu byl dále nainstalován správce balíčků npm (Node Package Manager). Npm je správce balíčků pro JavaScript a umožňuje jednoduché přidání a odebrání knihoven. Součástí inicializace projektu bylo také vytvoření Firebase konfigurace, která zajišťuje přístup k databázi Cloud Firestore a autentizaci uživatele. Na obrázku 10.1 je znázorněna struktura této konfigurace:

```
1 const firebaseConfig = {
2     apiKey: process.env.REACT_APP_FIREBASE_API_KEY,
3     authDomain: process.env.
          REACT_APP_FIREBASE_APP_DOMAIN+".firebaseapp.com
          ",
4     projectId: process.env.
          REACT_APP_FIREBASE_APP_DOMAIN,
5     storageBucket: process.env.
          REACT_APP_FIREBASE_APP_DOMAIN+".appspot.com",
6     appId: process.env.REACT_APP_FIREBASE_APP_ID
7 };
```

Obrázek 10.1: Příklad konfigurace Firebase

Adresářová struktura projektu

Na obrázku 10.2 je znázorněna adresářová struktura projektu vyjma některých konfiguračních souborů a obrázků. Ze struktury lze vidět, že veškeré zdrojové kódy jsou umístěny v jediném adresáři a následně roztříděny do logických sekcí.

```
cypress ..... end-to-end testy
public ..... výchozí HTML soubory aplikace
src ..... zdrojové kódy aplikace
    core ..... skripty pro tvorbu API requestů a další pomocné funkce
    redux ..... funkce pro práci s knihovnou Redux
    ui ..... UI prvky webové aplikace
        components ..... formulářové komponenty pro tvorbu reportů
        pages ..... navigační sekce webové aplikace
        structure ..... struktura webové aplikace
        styles ..... CSS soubory
    App.js ..... soubor obsahující logiku React aplikace
    index.js ..... výchozí JS soubor
Dockerfile ..... soubor Dockerfile
```

Obrázek 10.2: Adresářová struktura projektu

10.3.3 Autentizace a Firestore Security Rules

Firebase a Google Cloud služby nabízejí mnoho způsobů autentizace včetně přihlášení přes sociální sítě nebo telefonní číslo. V rámci této práce byla implementována standardní možnost autentizace s použitím emailu a hesla. Jedná se o jednoduché řešení a Firebase pro tuto variantu vystavuje uvedené metody:

- signInWithEmailAndPassword - přihlášení uživatele
- signOut - odhlášení uživatele

Dále byla na základě diagramu tříd (viz Příloha B) ve Firestore vytvořena databáze obsahující kolekci pro uložené konfigurace reportů. Přidanou hodnotou využití Google Cloud služeb pro autentizaci a přístup k datům je možnost použití Cloud Firestore Security Rules. Jedná se o nastavení oprávnění přístupu k datům, jenž umožňuje aby cizí uživatelé nemohli číst ani upravovat data jiných uživatelů. V kontextu této práce to přináší především možnost personalizace uložených reportů pro jednotlivé uživatele. Přidávat nové uživatele a spravovat Security Rules je možné v konzoli Firebase. Výchozí nastavení podporuje zápis dat pouze pro přihlášené uživatele. Nepřihlášený uživatel může pouze zobrazit již uložené konfigurace bez možnosti jejich úpravy nebo smazání.

10.3.4 Implementace komponent pro tvorbu reportů

Většina komponent frontendové aplikace byla vytvořena s použitím populární knihovny Reactstrap. Knihovna nabízí širokou škálu možností pro stylování a konfiguraci zejména formulářových prvků. Dále byla při implementaci využita knihovna Redux. Jedná se o knihovnu pro správu stavu aplikace (state management), jenž zajišťuje aktualizaci stavu výhradně prostřednictvím událostí. Hlavním důvodem pro její použití je možnost udržovat konzistentní stav

aplikace napříč více stránkami. Redux podporuje obecně lepší škálovatelnost aplikace [47]. V aplikaci slouží Redux například k udržení stavu načtených konfigurací reportů nebo dat o přihlášeném uživateli.

Další významnou částí implementace bylo využití knihoven pro tvorbu teplotních map na základě návrhu z předchozí kapitoly. Jedná se o knihovny BpmnJS a HeatmapJS. K zobrazení diagramu s BpmnJS je nejprve nutné získat XML soubor z Camunda REST API. Následně je na BPMN diagram vykreslena teplotní mapa s procesními daty pomocí HeatmapJS. Ukázku heatmap reportu lze najít v příloze D.2.

10.3.5 Ukázka získávání dat a tvorby reportu

Na obrázku 10.3 je ukázka zdrojového kódu, který řeší získávání dat pro tvorbu Ganttova diagramu. Vstupem pro vytvoření reportu je identifikátor procesu, který je zadán uživatelem. Na základě identifikátoru je následně pomocí REST API získána instance procesu a seznam dokončených aktivit. Následujícími voláními je získána definice procesu a XML soubor. Ze souboru jsou poté s využitím tagů získány všechny uživatelské úkoly. Tato ukázka slouží především k ilustraci využití Camunda REST API pro získávání dat uvnitř aplikace. Popsaný princip je obdobný i u ostatních druhů reportu.

```
1  getProcessInstanceById(processInstanceId)
2  .then(processInstance => {
3      if (!processInstance || !processInstance.state) {
4          return; }
5      getListOfFinishedActivityInstancesByProcessId(
           processInstanceId)
6      .then((activityInstances) => {
7          const currentActivityId = activityInstances.
               filter(...)
8          getCurrentProcessDefinitionIdByName(
               processDefinitionName)
9          .then(processDefinitionId => {
10             axios.get(`${baseUrl}/engine-rest/process-
                   definition/${processDefinitionId}/xml`)
11             .then(Api.parseAxios)
12             .then((data) => {
13                 const parser = new DOMParser();
14                 const xmlDoc = parser.parseFromString(
                       data.bpmn20Xml, "text/xml");
15                 const userTasks = xmlDoc.
                       getElementsByTagName("bpmn:userTask
                       ");
16                 const tasks = [];
17                 for (let i=0;i<userTasks.length;i++) {
18                 // add tasks to create Gantt Chart
19                 }
20 <Gantt tasks={tasks} .../>
```

Obrázek 10.3: Ukázka průběhu tvorby Ganttova diagramu

10.4 Nasazení a CI/CD

Architektura aplikace byla koncipována jako microservices, přičemž ve skutečnosti se opravdu jedná pouze o dvě samostatné služby pro backendovou a frontendovou část řešení (viz 8.2). Pro obě služby byl vytvořen soubor Dockerfile. Tento soubor obsahuje všechny potřebné specifikace k vytvoření přenosného kontejneru, který může být následně nasazen a spuštěn v libovolném prostředí [48]. Po nasazení služeb v rámci jednoho Docker prostředí komunikuje frontendová služba s backendovou službou pomocí síťového rozhraní hostitelského stroje na portu 8080, na který je backendová služba přesměrována v rámci Docker kontejneru.

Původním cílem bylo nasadit oba Docker kontejnery jako mikroslužby v prostředí Google Cloud již v rámci vývoje. V průběhu implementace bylo však zvoleno nasazení pouze backendové služby na platformě Google Cloud Run, zatímco frontendová služba byla nakonec nasazena na GitLab Pages. Důvodem byla především komerční povaha Google Cloud Platform a s tím spojené vysoké náklady. Dále bylo z praktických důvodů považováno za příjemnější a rychlejší nasazovat během vývojové fáze frontendovou službu na GitLab Pages, které umožňují přímé nasazení webových aplikací z fakultního GitLab repozitáře. Při každém odeslání do repozitáře dochází k automatickému nasazení nové verze aplikace. Veškerá konfigurace nasazení aplikace a přístupu k proměnným prostředí je možná v souboru `.gitlab-ci.yml`.

10.5 Závěr kapitoly

Tato kapitola měla za úkol přiblížit proces vývoje aplikace. Screenshoty z vytvořené aplikace je možné najít v příloze D. Aplikace je veřejně vystavena na adrese `https://forgaada.pages.fel.cvut.cz/camunda-reporting-tool/report`. Zdrojové kódy k backendové službě Camunda BPM jsou k nalezení v GitLab repozitáři s adresou `https://gitlab.fel.cvut.cz/forgaada/camunda-demo-project`. Zdrojové soubory ke klientské části aplikace jsou dostupné z `https://gitlab.fel.cvut.cz/forgaada/camunda-reporting-tool`.

Kapitola 11

Testování

Nezbytnou součástí tvorby softwaru je jeho testování. Tato kapitola je zaměřená na popis dvou hlavních přístupů, které byly zvoleny k testování aplikace. Prvním přístupem bylo kvalitativní uživatelské testování, které slouží k ověření použitelnosti a kontrole základních funkcionalit aplikace. Druhým přístupem bylo end-to-end testování pomocí nástroje Cypress. V této kapitole je rovněž zmíněna příprava testovacích procesů, které byly nasazeny v Camunda Platform 7.

11.1 Příprava testovacích procesů

V souvislosti se sběrem požadavků v rámci CZM (viz 5.3) bylo zvažováno nasazení reálných procesů do Camunda Platform a testování tvorby reportů nad statistickými daty těchto procesů. Bohužel vzhledem ke komplexnosti těchto procesů a citlivé povaze dat, se kterými pracují, byla tato možnost vyloučena. Pro účely ověření správného fungování aplikace byla tedy připravena sada celkem tří testovacích BPMN procesů:

- Proces objednávky
- Proces reklamace
- Proces tvorby ticketu

Jedná se o jednoduché procesy s několika aktivitami a přechody mezi nimi. Procesy byly namodelovány v Camunda Modeler (viz 4.2.1) a lze na nich ilustrovat možnosti tvorby reportů na základě statistických dat. Pro možnost generování Ganttova diagramu obsahují všechny úkoly těchto procesů klíčové atributy `camunda:dueDate` a `camunda:expectedDuration`. Soubory s BPMN diagramy zmíněných procesů jsou součástí backendové služby Camunda BPM a nacházejí se v adresáři `resources`. Procesy se nasazují automaticky při spuštění aplikace.

11.2 Kvalitativní uživatelské testování

V této sekci jsou popsány výstupy kvalitativního uživatelského testování aplikace, které probíhalo souběžně s vývojovou fází.

11.2.1 Průběh uživatelského testování

Za účelem testování byla oslovena skupina celkem deseti jedinců, z nichž někteří již měli zkušenosti s testováním softwaru i prací s platformou Camunda. Uživatelé byli nejprve seznámeni s používáním aplikace. Poté jim byly přiděleny přístupové údaje, aby si vyzkoušeli tvorbu a ukládání reportů nad nasazenými testovacími procesy. Jedním z hlavních účelů tohoto testování bylo ověřit, zda je aplikace dostatečně intuitivní pro uživatele, kteří mají základní povědomí o jejím využití a pohybují se v oblasti BPM.

Každému uživateli byl pro účely zpětné vazby zaslán elektronický dotazník. Jeho obsahem byl seznam několika otázek, které se týkaly jak celkové spokojenosti s používáním aplikace jako celku, tak spokojenosti s jednotlivými komponentami UI. Stěžejní výstupy uživatelského testování jsou uvedeny zde:

- Není možná filtrace instancí dle procesních proměnných.
- Navigační panel na levé straně obrazovky nereaguje na změnu URL či opětovné načtení stránky.
- Nepřihlášený uživatel může ukládat konfigurace reportu ale nemůže je upravovat a mazat.
- Okna s upozorněním zůstávají otevřená příliš dlouho a je nutné je zavírat manuálně.
- Názvy aktivit v procesním diagramu nejsou čitelné po vykreslení heatmapy.
- Filtrace dle procesních proměnných vrací i instance, které již neobsahují dané proměnné a jsou ve stavu "COMPLETED".
- Po vytvoření reportu a navigaci v prohlížeči zpět a vpřed není vidět dříve vytvořený report.
- U teplotní mapy není uvedena legenda, která by popisovala zobrazené barvy teplotní mapy a k nim příslušné hodnoty.

11.2.2 Vyhodnocení uživatelského testování

Na základě zpětné vazby lze usoudit, že aplikace byla přijata pozitivně. Uživatelé měli spíše pouze menší výhrady k některým nedostatkům UX. Většina těchto připomínek byla ihned zpracována. Zaměstnanci CZM, kteří aplikaci testovali, byli převážně potěšeni možnostmi, které aplikace přináší. Velmi dobrou zpětnou vazbu zaznamenala například možnost uložení reportů

pro pozdější využití. Tuto funkcionalitu mohou využívat analytici, kteří v současné době manuálně a periodicky tvoří některé druhy reportů.

Z bezpečnostních důvodů nebylo bohužel možné testovat klientskou aplikaci proti produkčnímu prostředí CZM. Přestože by ověření tvorby reportů nad reálnými procesními daty mělo obrovskou přidanou hodnotu, nízká úroveň zabezpečení aplikace a citlivost procesních dat představovali v tomto ohledu příliš vysoké riziko. Testování takto nabídlo uživatelům alespoň ukázku toho, k čemu může aplikace v praxi sloužit a otevírá možnost integrace systému v rámci CZM do budoucna.

11.3 End-to-end testování aplikace

Druhým zvoleným způsobem testování aplikace bylo end-to-end testování s použitím knihovny Cypress. Hlavní výhodou tohoto nástroje je jeho snadné použití a strmá křivka učení. Nástroj poskytuje rychlé ověření základních funkcionalit systému. V následující sekci je Cypress stručně představen a je uvedeno, jak byl v práci využit.

11.3.1 Cypress

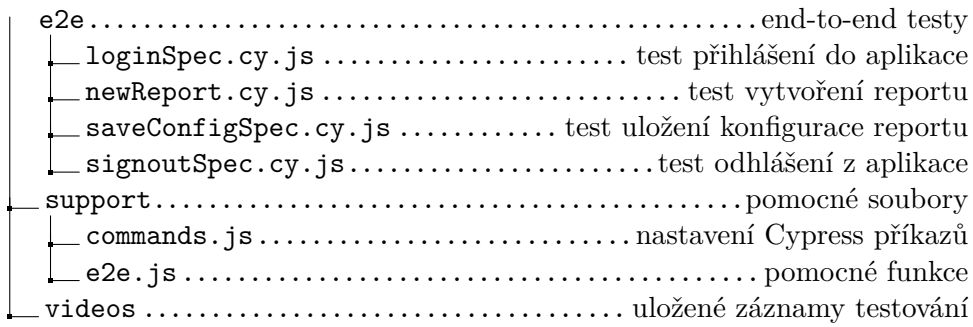
Cypress je populární knihovna pro testování klientských aplikací. Na rozdíl od většiny nástrojů pro end-to-end testování umožňuje Cypress rovněž testování komponent nebo tvorbu integračních testů. Hlavní výhodou je, že kombinuje několik nástrojů do jednoho a snižuje tak nároky na zkušenost vývojáře. V porovnání s většinou ostatních nástrojů je psaní testů za pomoci Cypress spolehlivější, snazší a rychlejší [49].

11.3.2 Tvorba testů

Cypress testy pokrývají základní funkcionalitu aplikace. Byly vytvořeny jednotlivé testy pro ověření:

- Přihlášení do aplikace
- Odhlášení z aplikace
- Vytvoření nového reportu
- Uložení konfigurace reportu

Cypress je možné spustit lokálně v headless módu pomocí příkazu `npx cypress run`. Ke spuštění Cypressu v GUI režimu slouží příkaz `npx cypress open`. Cypress také umožňuje ukládat záznamy z průběhu testování. Na obrázku 11.1 je vyobrazena adresářová struktura end-to-end Cypress testů.



Obrázek 11.1: Adresářová struktura Cypress end-to-end testů

11.4 Závěr kapitoly

Kapitola představila způsoby, jakými byla otestována nově vzniklá aplikace. Uživatelské testování bylo velice přínosné a ověřilo použitelnost aplikace běžnými uživateli. Během uživatelského testování nebyly detekovány žádné kritické chyby týkající se základních funkcionalit aplikace. Byly však nalezeny chyby v UX, které bylo nutné opravit. Dále byly pro rychlé ověření správného fungování aplikace vytvořeny end-to-end testy pomocí nástroje Cypress. Soubor testů je součástí projektu klientské aplikace.

Kapitola 12
Závěr

Cílem této diplomové práce bylo analyzovat existující reportingové nástroje pro procesní systémy a navrhnout, implementovat a otestovat prototyp aplikace, která bude umožňovat generování reportů nad procesními daty open-source platformy Camunda. Práce obsahuje rešerši existujících populárních BPM platforem a nástrojů pro analýzu procesních dat. Zejména byl blíže představen procesní systém Camunda Platform 7 a analytický nástroj Camunda Optimize. Z průzkumu vyplynulo, že pro analýzu procesních dat z platformy Camunda není k dispozici žádné volně dostupné řešení. Komerční nástroje, které jsou k dispozici, vyžadují placené licence a nejsou tak vhodné pro menší firmy nebo instituce. Camunda Platform však vystavuje veřejné rozhraní pro práci s procesními daty, což otevírá možnost tvorby vlastního nástroje.

Na základě rešerše existujících analytických nástrojů a dotazníku na zaměstnance Centra znalostního managementu FEL ČVUT byl proveden sběr požadavků na nově vytvářenou aplikaci reportingového nástroje. Byly definovány klíčové funkce, kterými by aplikace měla disponovat. Mezi požadované způsoby vizualizace statistických dat, které aplikace poskytuje, patří například tabulkové reporty, teplotní mapy, sloupcové grafy nebo Ganttův diagram. V rámci analytické části této práce byly rovněž zahrnuty kapitoly pokrývající funkční analýzu a analýzu Camunda REST API. V souladu s požadavky byla navržena architektura aplikace, zvoleny potřebné technologie pro vývoj a připraveny návrhy obrazovek. Architektura aplikace byla koncipována jako servisní architektura, přičemž aplikace se sestává z backendové SpringBoot služby a frontendové React.js aplikace.

Poslední část práce se zaměřuje na tvorbu prototypu aplikace. Samostatná kapitola byla věnována implementaci. V této kapitole je popsána inicializace projektu Camunda BPM a implementace frontendové části aplikace. Dále je vysvětleno, jak byly použity technologie Google Firebase a Cloud Firestore zajišťující autentizaci a práci s uloženými reporty. Pro backendovou i frontendovou službu byly vytvořeny Docker kontejnery, umožňující nasazení a spuštění v libovolném prostředí. Součástí vývojové fáze byla také tvorba testovacích BPMN procesů určených k ověření funkčnosti aplikace. Aplikace byla otestována prostřednictvím kvalitativního uživatelského testování a Cypress end-to-end testů.

Z výstupů uživatelského testování lze usoudit, že práce úspěšně splnila všechny body zadání. Uživatelé vznesli pouze menší připomínky týkající se použitelnosti a návrhu UX/UI. Většina těchto chyb byla v průběhu implementace opravena. Někteří uživatelé, kteří se testování zúčastnili, mají v úmyslu aplikaci používat i nadále, neboť jim výrazně usnadnila manuální analýzu dat.

12.1 Budoucnost aplikace

Realizací prototypu aplikace bylo ověřeno, že existuje možnost vytvoření vlastního řešení pro analýzu dat platformy Camunda. Nová aplikace je v aktuálním stavu plně funkční a její zdrojové kódy jsou volně dostupné v GitLab repozitáři na adrese `https://gitlab.fel.cvut.cz/forgaada/camunda-reporting-tool`. Uživatelé mohou aplikaci lokálně nainstalovat a integrovat do vlastních systémů, které pracují s platformou Camunda.

Přestože aplikace implementuje veškeré funkční požadavky, které byly definovány v rámci této práce, je zde stále spousta prostoru pro její vylepšení. V budoucnu by aplikace mohla být obohacena například o dashboard sekci, která by nahradila současnou sekci přehledu konfigurací reportů. Uživatel by mohl ihned po spuštění aplikace vidět miniatury uložených reportů. Tato funkcionalita nebyla doposud implementována z důvodu vyšší technické náročnosti.

Dále se nabízí rozšíření aplikace o dodatečné způsoby reportování dat. Ganttův diagram by mohl být rozšířen o zobrazení milníků projektu či komplexnější způsob vykreslování přechodů mezi úkoly. Existují také různé další druhy grafů a diagramů, které by bylo možné realizovat s využitím procesních dat platformy Camunda. Výhodou Camunda REST API je, že poskytuje různorodá data, jenž otevírají dveře pro vývoj mnoha dalších funkcionalit. V tomto ohledu mohou uživatelé, kteří aplikaci budou využívat, libovolně přidávat vlastní funkcionality, které budou odpovídat jejich vlastním potřebám a požadavkům na analýzu dat.

Přílohy

Příloha A

Literatura a zdroje

1. *Business Process* [online]. Edmonton, Alberta, Canada: Techopedia, 2015 [cit. 2023-05-16]. Dostupné z: https://www.techopedia.com/definition/1168/business-process.
2. DUMAS, Marlon; ROSA, Marcello La; MENDLING, Jan; REIJERS, Hajo A. *Fundamentals of business process management*. Druhé vydání. Berlin, Germany: Springer Berlin, 2018. ISBN 9783662565087.
3. *What is BPMS?* [online]. Washington, USA: Microsoft Corporation, 2022 [cit. 2023-05-16]. Dostupné z: https://powerautomate.microsoft.com/en-us/bpms-business-process-management-software/.
4. *What Is a BPMS? A Guide to Business Process Management Systems* [online]. Durham, USA: ProcessMaker Inc., 2022 [cit. 2023-05-16]. Dostupné z: https://www.processmaker.com/blog/what-is-a-bpms-a-guide-to-business-process-management-systems/.
5. SZELĄGOWSKI, Marek. Evolution of the BPM Lifecycle. *2018 Federated Conference on Computer Science and Information Systems*. 2018, s. 205–211. Dostupné z DOI: 10.15439/2018F46.
6. *Business Process Management Platforms Reviews and Ratings* [online]. Stamford, USA: Gartner, 2022 [cit. 2023-05-16]. Dostupné z: https://www.gartner.com/reviews/market/business-process-management-platforms.
7. *IBM Business Process Manager Overview* [online]. USA: IBM, 2021 [cit. 2023-05-16]. Dostupné z: https://www.ibm.com/docs/en/bpm/8.6.0?topic=manager-business-process-overview.
8. *Business Process Management* [online]. Walldorf, Německo: SAP, 2021 [cit. 2023-05-16]. Dostupné z: https://help.sap.com/docs/SAP_NETWEAVER_731/1e3334e30d8548038947e9792f6cb376/10e808e319284dc8b0eac42d1d95735a.html?version=7.31.25.
9. *Microsoft Power Automate* [online]. Washington, USA: Microsoft, 2022 [cit. 2023-05-16]. Dostupné z: https://powerautomate.microsoft.com/en-us/.
10. *Low-Code Process Automation Platform* [online]. USA: Bizagi, 2022 [cit. 2023-05-16]. Dostupné z: https://www.bizagi.com/en/platform.

11. *Patterns-based evaluation of open source BPM systems: The cases of jBPM, OpenWFE, and Enhydra Shark* [online]. Amsterdam, Nizozemsko: Elsevier, 2009 [cit. 2023-05-16]. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0950584909000263.

12. *12 of the Top-Rated Free and Open-Source BPM Software Solutions* [online]. USA: Solutions Review, 2022 [cit. 2023-05-16]. Dostupné z: https://solutionsreview.com/business-process-management/the-top-free-and-open-source-bpm-software/.

13. *Activiti* [online]. Activiti, 2023 [cit. 2023-05-16]. Dostupné z: https://www.activiti.org/.

14. *What is jBPM?* [online]. jBPM, 2023 [cit. 2023-05-16]. Dostupné z: https://www.jbpm.org/.

15. *Camunda Platform 7* [online]. USA: Camunda, 2022 [cit. 2023-05-16]. Dostupné z: https://camunda.com/platform-7/.

16. CORTES-CORNAX, Mario. *Business Information Systems : 20th International Conference, BIS 2017, Poznan, Poland, June 28-30, 2017, Proceedings*. První vydání. Berlin, Germany: Springer, 2017. ISBN 9783319593357.

17. *Without Excellent Reports, Your BPM Software is Uncooked* [online]. Chennai, India: Kissflow, 2021 [cit. 2023-05-16]. Dostupné z: https://kissflow.com/workflow/bpm/without-excellent-reports-your-bpm-software-is-uncooked/.

18. *What is Reporting Software?* [online]. Toronto, Canada: PAT Research, 2022 [cit. 2023-05-16]. Dostupné z: https://www.predictiveanalyticstoday.com/what-is-reporting-software/.

19. *What is Process Intelligence and 5 Reasons Why It Matters?* [online]. Tallinn, Estonia: AI Multiple, 2022 [cit. 2023-05-16]. Dostupné z: https://research.aimultiple.com/process-intelligence/.

20. *Analytics and Business Intelligence Platforms Reviews and Ratings* [online]. Stamford, USA: Gartner, 2022 [cit. 2023-05-16]. Dostupné z: https://www.gartner.com/reviews/market/analytics-business-intelligence-platforms.

21. *What is Power BI?* [online]. Washington, USA: Microsoft, 2022 [cit. 2023-05-16]. Dostupné z: https://learn.microsoft.com/cs-cz/power-bi/fundamentals/power-bi-overview.

22. *SAP Crystal Reports* [online]. Walldorf, Německo: SAP, 2022 [cit. 2023-05-16]. Dostupné z: https://www.sap.com/products/technology-platform/crystal-reports.html.

23. BHOMBEA, Aarjavi; WALUKARA, Kshitij; THAKAREB, Yash; KAMBLE, Shailesh. *Comparative Analysis of two BI tools: Micro Strategy and Tableau* [online]. Amsterdam, Nizozemsko: Elsevier, 2019 [cit. 2023-05-16]. Dostupné z: https://ssrn.com/abstract=3462539.

24. *Camunda Optimize* [online]. USA: Camunda, 2022 [cit. 2023-05-16]. Dostupné z: https://camunda.com/platform/optimize/.
25. *Camunda hauls in 28M investment as workflow automation remains hot* [online]. San Francisco, USA: Tech Crunch, 2018 [cit. 2023-05-16]. Dostupné z: https://techcrunch.com/2018/12/05/camunda-hauls-in-28m-investment-as-workflow-automation-remains-hot/.
26. *Free BPMN 2.0 tool* [online]. Berlin, Germany: Camunda, 2022 [cit. 2023-05-16]. Dostupné z: https://camunda.com/bpmn/tool/.
27. *Moving from Embedded to Remote Workflow Engines* [online]. Berlin, Germany: Camunda, 2022 [cit. 2023-05-16]. Dostupné z: https://camunda.com/blog/2022/02/moving-from-embedded-to-remote-workflow-engines/.
28. *The Camunda BPM Manual* [online]. USA: Camunda, 2022 [cit. 2023-05-16]. Dostupné z: https://docs.camunda.org/manual/7.5/introduction/.
29. *Introducing Camunda Platform 8* [online]. USA: Camunda, 2022 [cit. 2023-05-16]. Dostupné z: https://camunda.com/platform/.
30. *Enterprise Reporting Tools Features and Functional Requirements Checklist* [online]. Texas, USA: SelectHub, 2022 [cit. 2023-05-16]. Dostupné z: https://www.selecthub.com/business-intelligence/enterprise-reporting/reporting-tools-features-requirements/.
31. GU, Zuguang. Complex heatmap visualization. *iMeta*. 2022, roč. 1, č. 3. Dostupné z DOI: 10.1002/imt2.43.
32. *What is Gantt chart?* [online]. Walnut, CA: ProofHub, 2023 [cit. 2023-05-16]. Dostupné z: https://www.proofhub.com/articles/gantt-charts.
33. PROJECT MANAGEMENT INSTITUTE, Inc. *The standard for project management and a guide to the project management body of knowledge: PMBOK GUIDE*. Seventh. Newton Square, Pennsylvania: Project Management Institute, Inc, 2021. ISBN 9781628256642.
34. *What Are Functional Requirements? Types and Examples* [online]. Estonia: WinATalent, 2021 [cit. 2023-05-16]. Dostupné z: https://winatalent.com/blog/2020/05/what-are-functional-requirements-types-and-examples/.
35. *Functional and Nonfunctional Requirements of Software* [online]. Ukraine: GBKSOFT, 2021 [cit. 2023-05-16]. Dostupné z: https://gbksoft.com/blog/functional-and-nonfunctional-requirements-the-detailed-guide/.
36. STEPHENS, Rod. *Beginning Software Engineering*. První vydání. Birmingham, UK: Wrox, 2015. ISBN 9788126555376.
37. LARRUCEA, Xabier; SANTAMARIA, Izaskun; COLOMO-PALACIOS, Ricardo; EBERT, Christof. Microservices. *IEEE Software*. 2018, roč. 35, č. 3, s. 96–100. Dostupné z DOI: 10.1109/MS.2018.2141030.

38. *Cloud Firestore* [online]. USA: Google, 2023 [cit. 2023-05-16]. Dostupné z: `https://firebase.google.com/docs/firestore`.

39. ARORA, Chandermani; HENNESSY, Kevin. *Angular 6 by Example: Get up and Running with Angular by Building Modern Real-World Web Apps, 3rd Edition*. Třetí vydání. UK: Packt Publishing, Limited, 2018. ISBN 9781788835176.

40. HALLIDAY, Paul. *Vue. js 2 Design Patterns and Best Practices: Build Enterprise-Ready, Modular Vue. js Applications with Vuex and Nuxt*. První vydání. UK: Packt Publishing, Limited, 2018. ISBN 9781788839792.

41. ROLDÁN, Carlos Santana. *React Cookbook: Create Dynamic Web Apps with React Using Redux, Webpack, Node. js, and GraphQL*. První vydání. UK: Packt Publishing, Limited, 2018. ISBN 9781783980727.

42. KELONYE, Mitchel. *Mastering Ember.js*. První vydání. UK: Packt Publishing, Limited, 2014. ISBN 9781783981984.

43. *Moqups* [online]. Romania: S.C Evercoder Software S.R.L., 2021 [cit. 2023-05-16]. Dostupné z: `https://moqups.com/`.

44. *Camunda Automation Platform 7 Initializr* [online]. USA: Camunda, 2023 [cit. 2023-05-16]. Dostupné z: `https://start.camunda.com/`.

45. *WebStorm* [online]. Česká Republika: JetBrains s.r.o., 2023 [cit. 2023-05-16]. Dostupné z: `https://www.jetbrains.com/webstorm/`.

46. *Create React App* [online]. USA: Facebook, Inc., 2023 [cit. 2023-05-16]. Dostupné z: `https://create-react-app.dev/`.

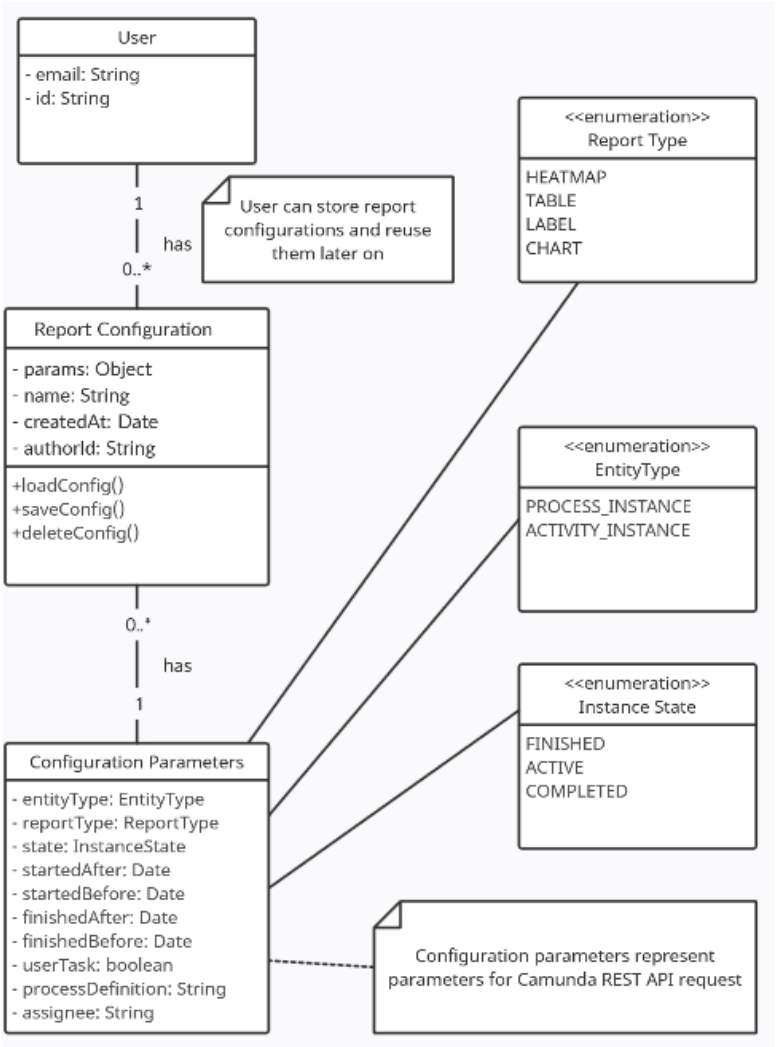
47. THAKKAR, Mohit. *Building React Apps with Server-Side Rendering: Use React, Redux, and Next to Build Full Server-Side Rendering Applications*. První vydání. Berkeley, CA: Apress, 2020. ISBN 9781484258682.

48. *Docker* [online]. Palo Alto, CA: Docker, Inc., 2023 [cit. 2023-05-16]. Dostupné z: `https://www.docker.com/`.

49. *Why Cypress?* [online]. Atlanta, Georgia: Cypress.io, 2023 [cit. 2023-05-16]. Dostupné z: `https://docs.cypress.io/guides/overview/why-cypress`.

Příloha B

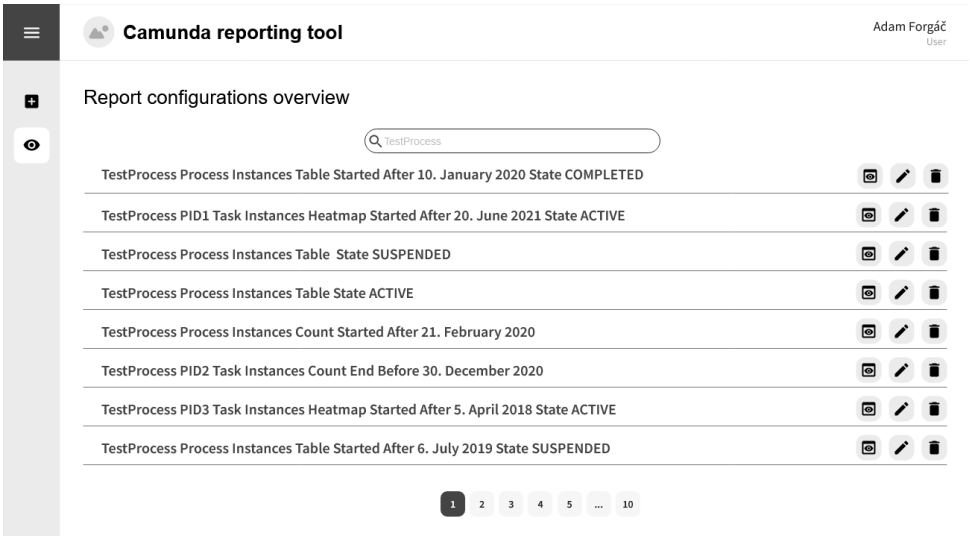
Diagram tříd



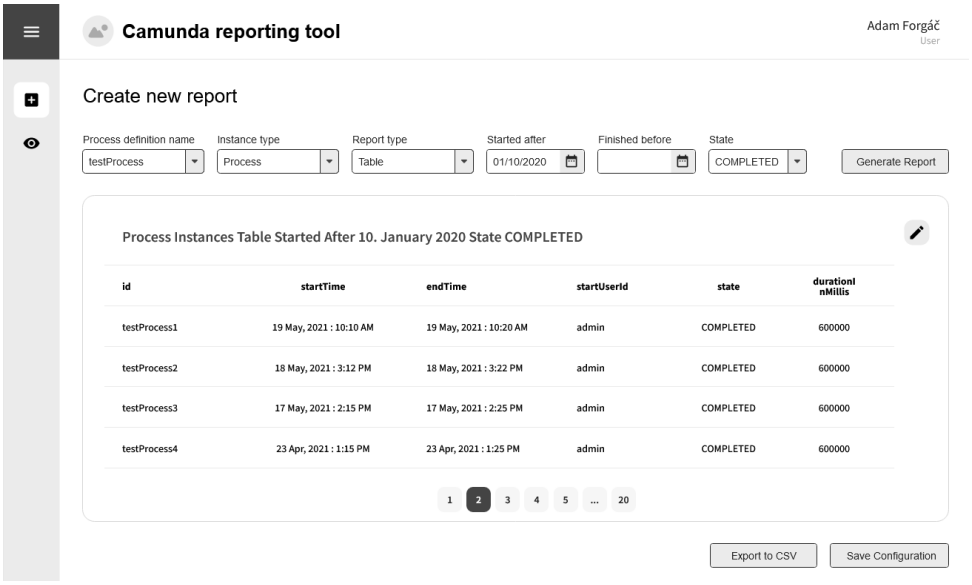
Obrázek B.1: Diagram tříd

Příloha C

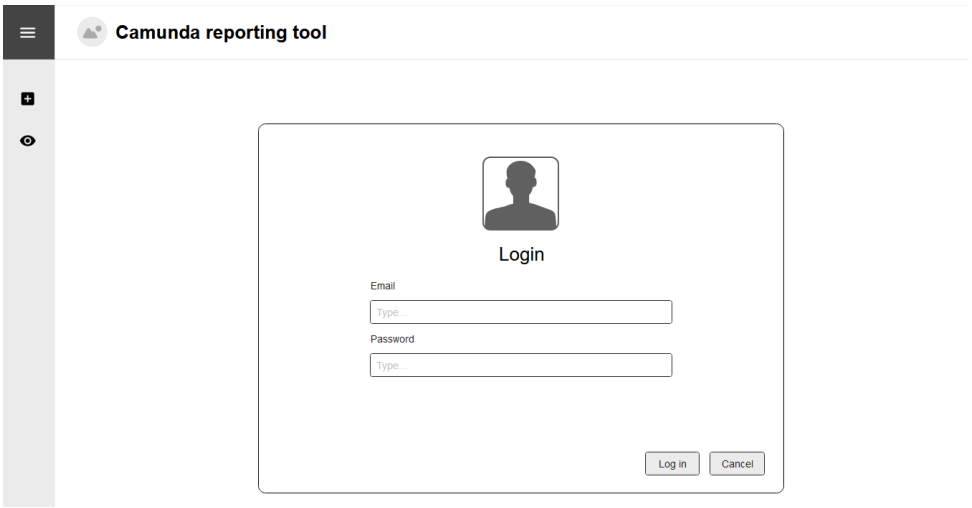
Ukázky návrhů obrazovek



Obrázek C.1: Návrhy obrazovek – Přehled reportů



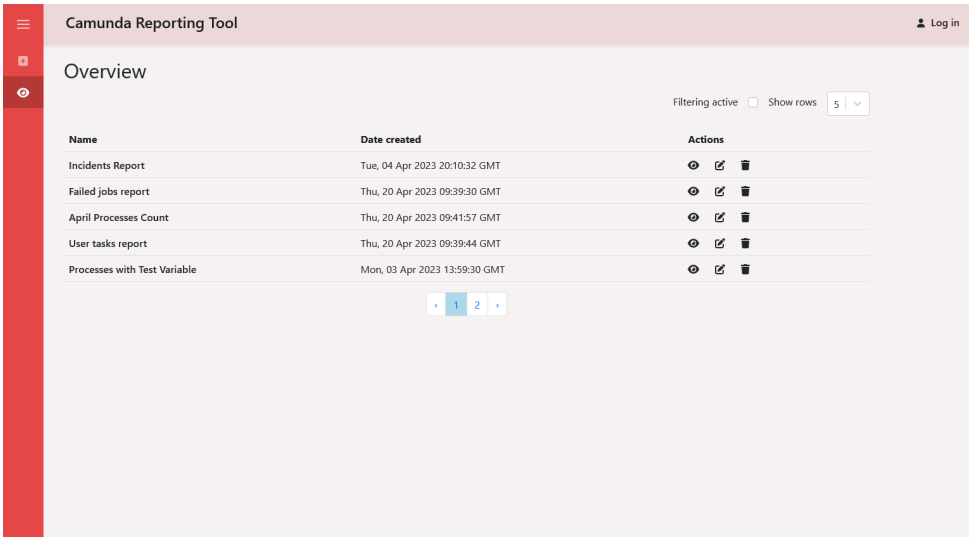
Obrázek C.2: Návrhy obrazovek – Tvorba nového reportu



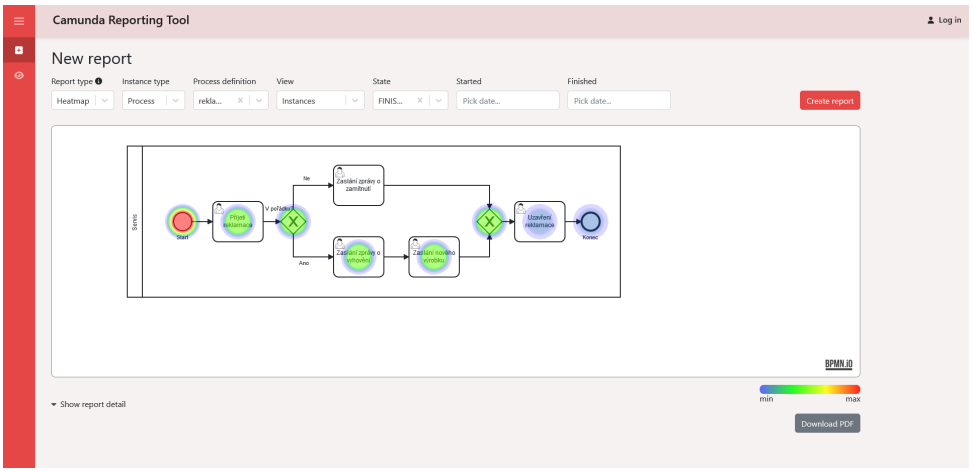
Obrázek C.3: Návrhy obrazovek – Přihlášení

Příloha D

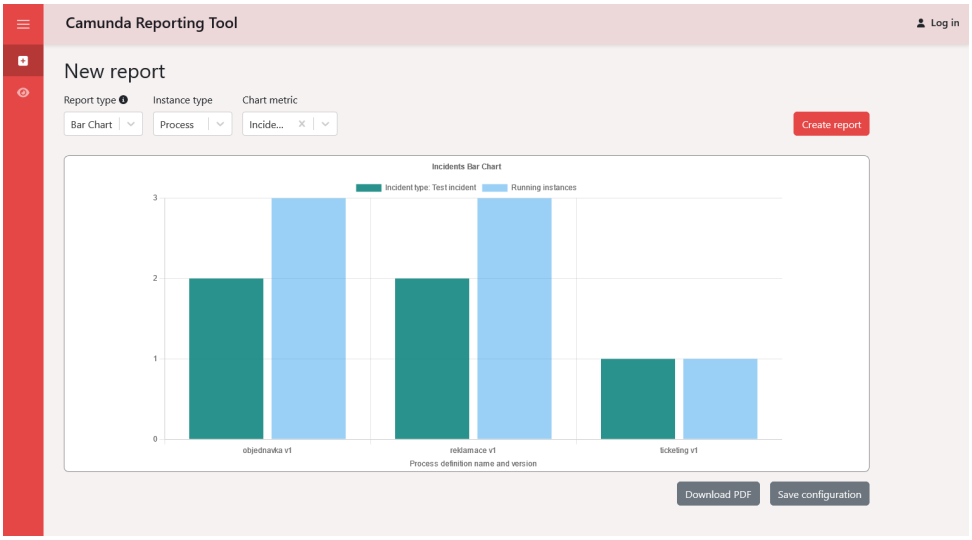
Screenshoty z klientské aplikace



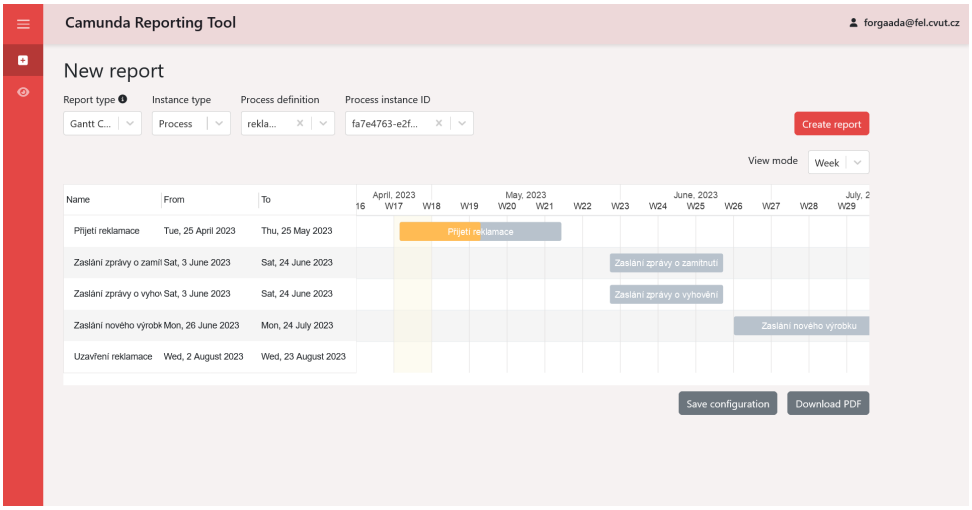
Obrázek D.1: Snímek obrazovky – Přehled uložených konfigurací reportu



Obrázek D.2: Snímek obrazovky – Heatmap report



Obrázek D.3: Snímek obrazovky – Incident report



Obrázek D.4: Snímek obrazovky – Gantt report

Příloha E

Seznam použitých zkratek

API	Application Programming Interface
BI	Business Intelligence
BPI	Business Process Intelligence
BPM	Business Process Management
BPMN	Business Process Model and Notation
BPMS	Business Process Management Software
CMMN	Case Management Model and Notation
CSS	Cascading Style Sheets
CSV	Comma-separated values
CZM	Centrum znalostního managementu na FEL ČVUT
DMN	Decision Model and Notation
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
LDAP	Lightweight Directory Access Protocol
NPM	Node Package Manager
PDF	Portable Document Format
PI	Process Intelligence
REST	Representational State Transfer
SaaS	Software as a Service
SPA	Single-page application
XML	Extensible Markup Language